Lemmas: Generation, Selection, Application

Michael Rawson [1] Christoph Wernhard [2] Zsolt Zombori [3] Wolfgang Bibel [4]

[1]TU Wien [2]University of Potsdam [3]Alfréd Rényi Institute of Mathematics and Eötvös Loránd University
[4]Technical University Darmstadt

TABLEAUX 2023

Prague, Czech Republic, Sep 18–21, 2023

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 457292495, by the North-German Supercomputing Alliance (HLRN), by the ERC grant CoG ARTIST 101002685, by the Hungarian National Excellence Grant 2018-1.2.1-NKP-00008, the Hungarian Artificial Intelligence National Laboratory Program (RRF-2.3.1-21-2022-00004), the ELTE TKP 2021-NKTA-62 funding scheme and the COST action CA20111.

PROJECT FINANCED FROM THE NRDI FUND

Lemmas: Generation, Selection, Application

1. Introduction: Learning Useful Lemmas
2. A Framework that Incorporates Proof Structures
3. Experiments: Improving a Prover via Learned Lemma Selection
4. Experiment: Proving LCL073-1 with Lemmas
5. Conclusion

Lemmas in Mathematics

- May help to find a proof more easily
- Can be applied several times, but need to be proven only once
- Can help to structure a proof for human comprehension

Lemmas in ATP

- In general **factorize duplication**, e.g., of subproofs within a proof or among different proofs
- Play a different role, depending on the prover family
  - **Provers that internally maintain lemmas**: A resolvent is a lemma that can be re-used
  - **Provers without internal lemmas**: Connection Method / Clausal tableaux (**“CM-CT”**) provers perform top-down proof search from the goal where subgoals are proven repeatedly
- Can be applied as **external input lemmas** in different ways
  - **Adding** the lemmas to the original axioms
    - shortens proofs
    - widens search possibilites
  - **Replacing** parts of the search by lemma access
    - alters, restricts the overall search
- Ideally, for a given problem we would like to **identify just a few relevant lemmas**

Learning Useful Lemmas

- **Learning the utility of lemmas**
  Does a lemma move the goal closer to the axioms?
- [Kaliszyk, Urban 2015]: identify globally useful lemmas in millions of HOL Light proofs
- Here: evaluating lemmas **in the context of an axiom set and a goal**
- Like premise selection, but no given premise set: **generate, select, apply lemmas**
- MaLARea [Urban et al. 2008]: **iterative improvement**

Condensed Detachment (CD)

- By Carew A. Meredith (1904–1976) – mid 1950s
- A **D-term (full binary tree)** proves for given axioms its **most general theorem (MGT)**, determined by **unification**
- A possible inference system for CD

$$\frac{}{1 : \mathsf{P}(t)}\,\textit{fresh-copy} \qquad \text{for axiom } \mathsf{P}(t)$$

$$\frac{d_1 : \mathsf{P}(\mathsf{i}(x,y)) \qquad d_2 : \mathsf{P}(x')}{\mathsf{D}(d_1, d_2) : \mathsf{P}(y)}\,mgu(x, x')$$

- CD problems as first-order ATP problems

Detachment axiom	$\mathsf{P}(\mathsf{i}(x,y)) \land \mathsf{P}(x) \rightarrow \mathsf{P}(y)$
Proper axioms	positive units, e.g. $\mathsf{P}(\mathsf{i}(x, \mathsf{i}(y, x)))$
Goal	negative ground unit, e.g. $\neg\mathsf{P}(\mathsf{i}(\mathsf{a}, \mathsf{a}))$

Horn, first-order, binary function symbol, cyclic predicate dependency

- Relation to CM and more: [CW, Bibel CADE 21; 2023]

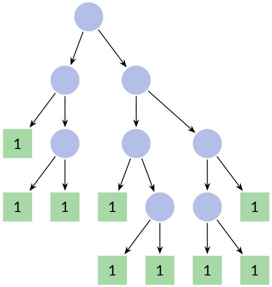
1. *CCCpqrCCrpCsp*
2. *CCCpqpCrp* = DDD1D111n
3. *CCCpqrCqr* = DDD1D1D121n
4. *CpCCpqCrq* = D31
5. *CCCpqCrsCCCqtsCrs* = DDD1D1D1D141n
6. *CCCpqCrsCCpsCrs* = D51
7. *CCpCqrCCpsrCqr* = D64
8. *CCCCCpqrtCspCCrpCsp* = D71
9. *CCpqCpq* = D83
10. *CCCCrpCtpCCCpqrsCuCCCpqrs* = D18
11. *CCCCpqrCsqCCCqtsCpq* = DD10.10.n
12. *CCCCpqrCsqCCCqtpCsq* = D 5.11
13. *CCCCpqrsCCsqCpq* = D12.6
14. *CCCpqrCCrpp* = D12.9
15. *CpCCpqq* = D3.14
16. *CCpqCCCprqq* = D6.15

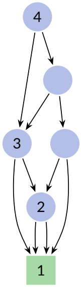
*17. *CCpqCCqrCpr* = DD.13D.16.16.13

*18. *CCCpqpp* = D14.9

*19. *CpCqp* = D33

Size Measures for D-Terms (Full Binary Trees)





Term representation

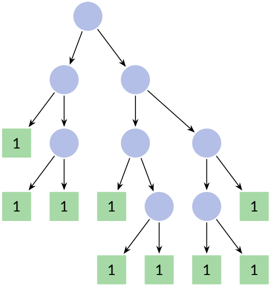
$\mathsf{D}(\mathsf{D}(1, \mathsf{D}(1, 1)), \mathsf{D}(1, \mathsf{D}(\mathsf{D}(1, 1)), \mathsf{D}(\mathsf{D}(1, 1), 1)))$

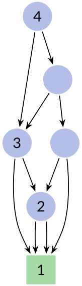
Representation by factor equations

$$\begin{array}{rcl} 2 & = & \mathsf{D}(1, 1) \\ 3 & = & \mathsf{D}(1, 2) \\ 4 & = & \mathsf{D}(3, \mathsf{D}(3, \mathsf{D}(2, 1))) \end{array}$$

- **Tree size**: 8
- **Height**: 4
- **Compacted size**: 5 – size of minimal DAG; number of distinct compound subterms

D-Terms and Lemmas





- **Proven unit lemma = D-term (tree) with its MGT**
- A subterm of a D-term also represents such a lemma
- The DAG view expresses lemma re-use
- Features of both D-term and MGT are available for learning and selecting
- Lemma generation: enumerating D-terms with MGT
- **Enumerating D-terms is also an ATP approach**, generalizing the enumeration of proof structures underlying CM-CT provers
- Enumeration can be performed upon increasing levels, e.g. tree size or height of the D-terms

SGCD – Structure-Generating Theorem Proving for CD

- Assume a Prolog predicate that, depending on the parameter instantiation, serves different purposes

`enum_dterm_mgt_pairs`$(+Level, +Dterm, +Formula)$	verifying a proof
`enum_dterm_mgt_pairs`$(+Level, +Dterm, -Formula)$	computing the MGT
`enum_dterm_mgt_pairs`$(+Level, -Dterm, +Formula)$	**proving a formula (goal-driven)**
`enum_dterm_mgt_pairs`$(+Level, -Dterm, -Formula)$	**generating lemmas (axiom-driven)**

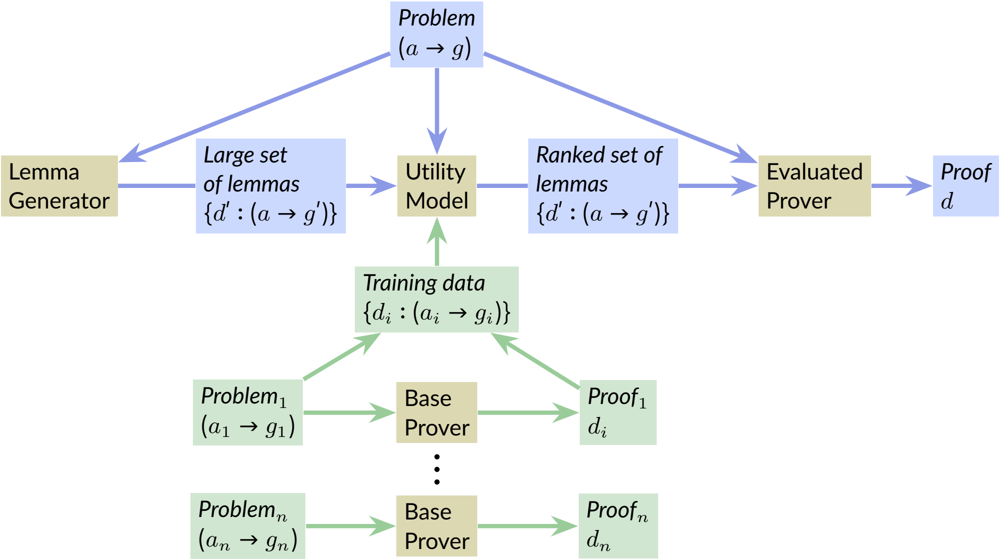
- SGCD embeds it in **nested loops of goal- and axiom-driven phases**
- A **cache** collects the results of the axiom-driven phases
- Subproblems for lower levels are **solved from the cache**
- The cache can be **heuristically restricted on the basis of MGTs**
- Optional: **replacing lemma application** – initializing the cache with given lemmas
- Optional: "hybrid levels": different level characterizations for goal- and axiom-driven

$Cache$:= ∅;
for $l := 0$ **to** $maxLevel$ **do**
  for $m := l$ **to** $l + preAddMaxLevel$ **do**
    `enum_dterm_mgt_pairs`$(m, d, goal)$;
    throw `proof_found`(d)
  $N := \{\langle l, d, f\rangle \mid$ `enum_dterm_mgt_pairs`$(l, d, f)\}$;
  if $N = \emptyset$ **then throw** `exhausted`;
  $Cache$:= `merge_news_into_cache`$(N, Cache)$

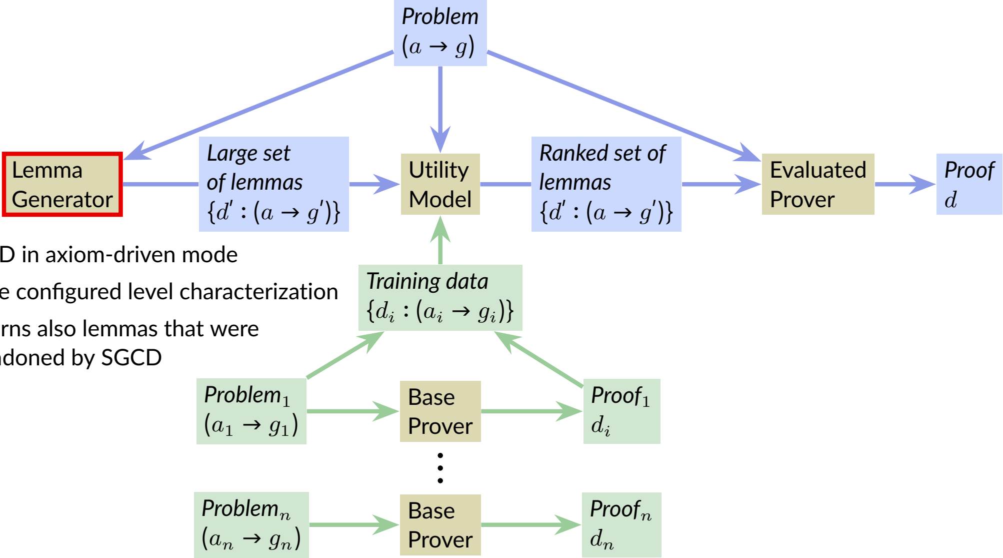
312 CD problems

- The 196 “pure” CD problems in the TPTP
  (all CD problems in the TPTP except 10 with: status *satisfiable*; detachment with disj. and neg.; goal theorem not an atom)
- Single-axiom versions of 116 multi-axiom problems in these 196, obtained with the “Tarski/Rezuş technique” [Rezuş 2010]
- No split into training and test problems

Method Overview

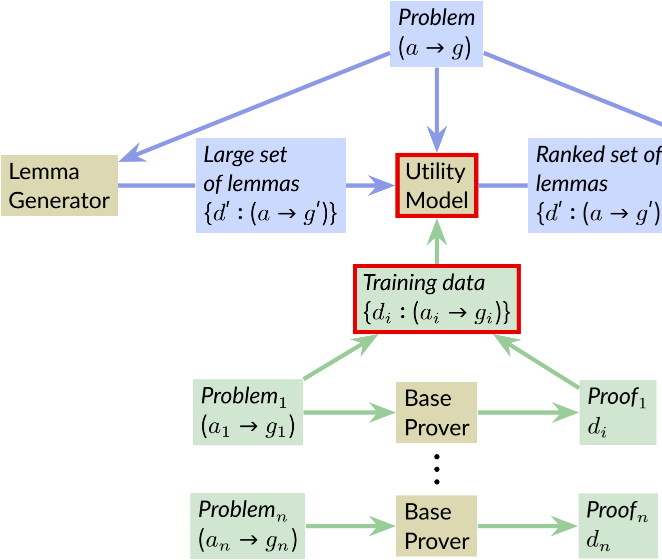


Lemma Generation



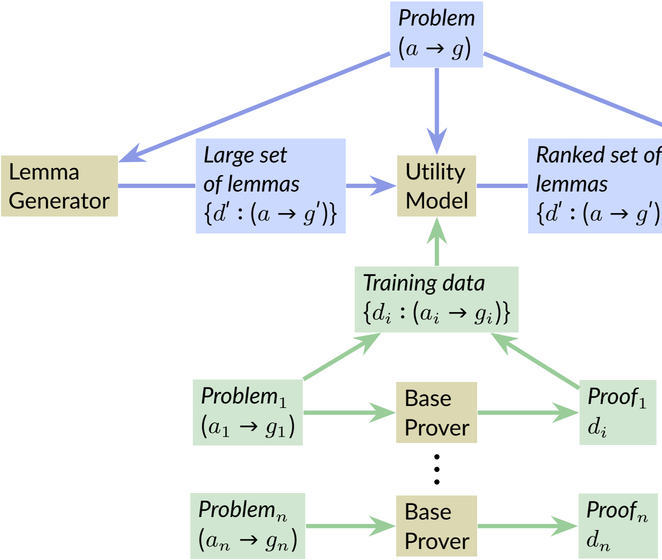
- SGCD in axiom-driven mode
- Some configured level characterization
- Returns also lemmas that were abandoned by SGCD

The Utility Model



- Uses machine learning
- 60 manual or automated (Graph Neural Network – GNN) input features
- Features consider D-term and MGT of the lemma, axioms, goal
- Training utility of a lemma based on proving-time reduction
- Generated lemmas ranked according to utility score $u \in [0, 1]$

Lemma Application



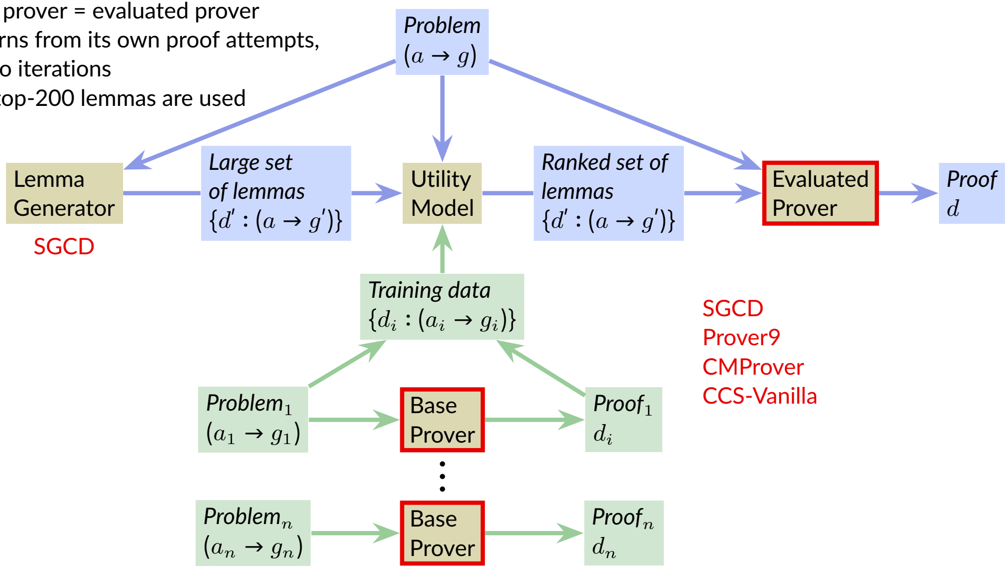
- Select the top k lemmas
- Optionally: enrich them by closure under sub-proofs
- Add the lemmas as axioms, or, if the prover supports it, in a replacing mode

Considered Provers

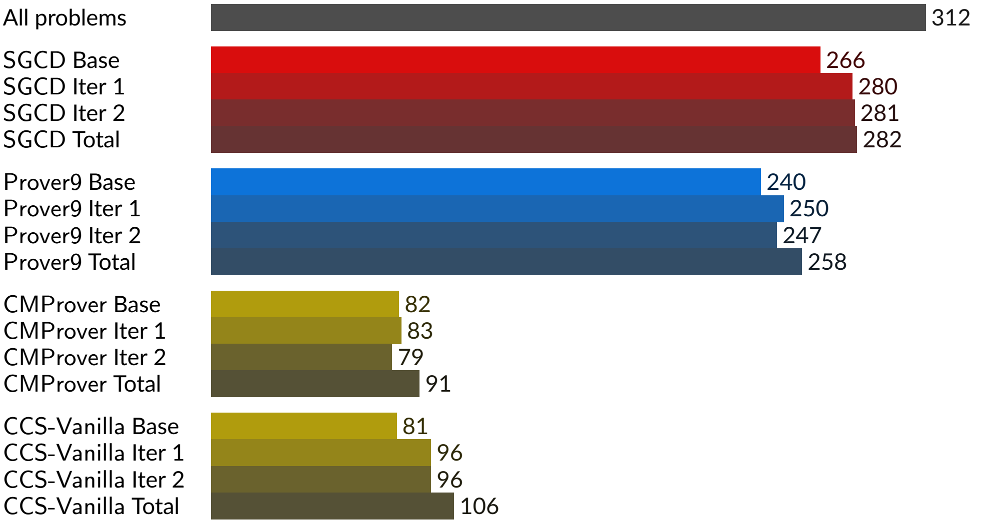
	SGCD	Prover9	CMProver	leanCoP	CCS-Vanilla	Vampire	E
Internal lemmas	✓	✓				✓	✓
External lemmas that replace search	✓				✓		
Outputs D-terms: allows use for training	✓	✓	✓		✓		

Experiment 1: Iterative Improvement of the Base Prover

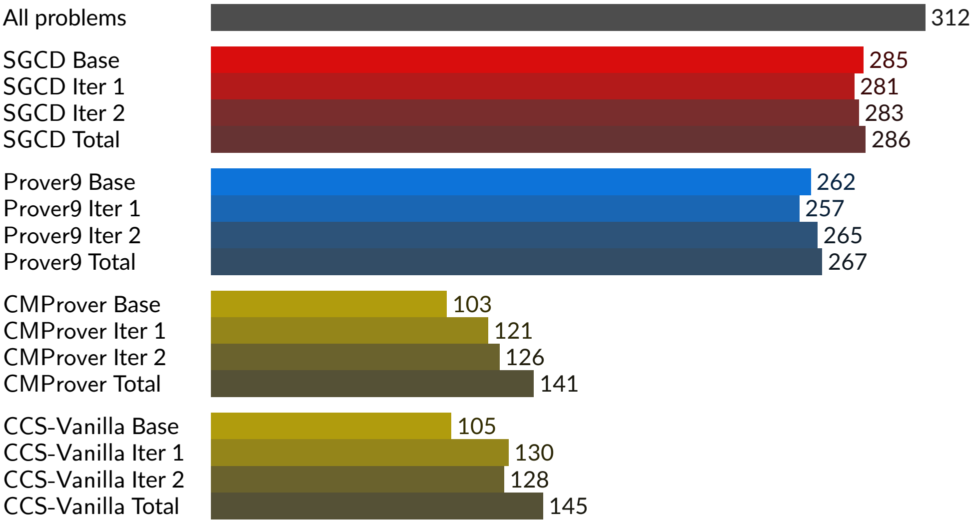
- Base prover = evaluated prover
- It learns from its own proof attempts, in two iterations
- The top-200 lemmas are used



Experiment 1: Iterative Improvement of the Base Prover – Results for Time limit 50 s

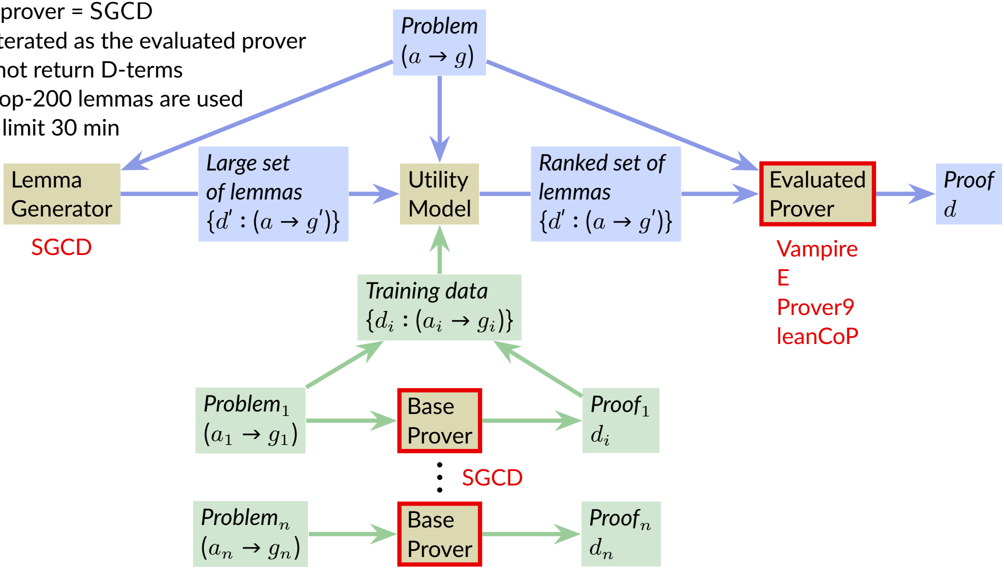


Experiment 1: Iterative Improvement of the Base Prover – Results for Time limit 30 min

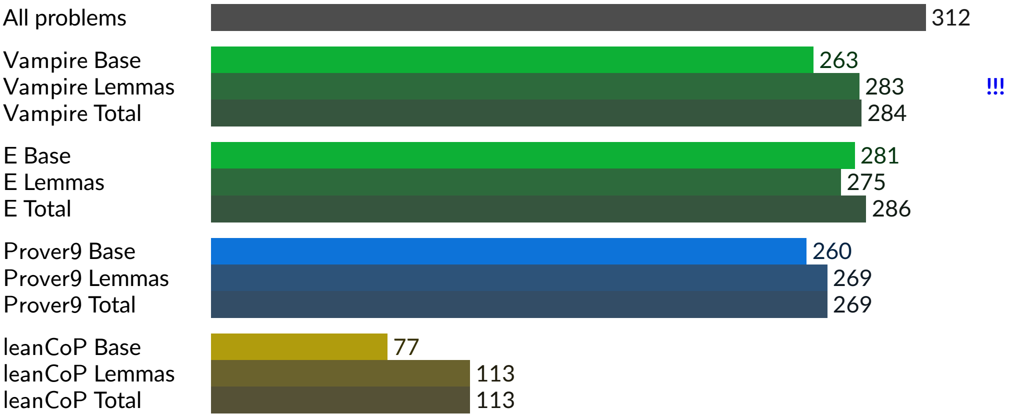


Experiment 2: Learned Lemmas to Enhance Other Provers

- Base prover = SGCD
- Not iterated as the evaluated prover may not return D-terms
- The top-200 lemmas are used
- Time limit 30 min

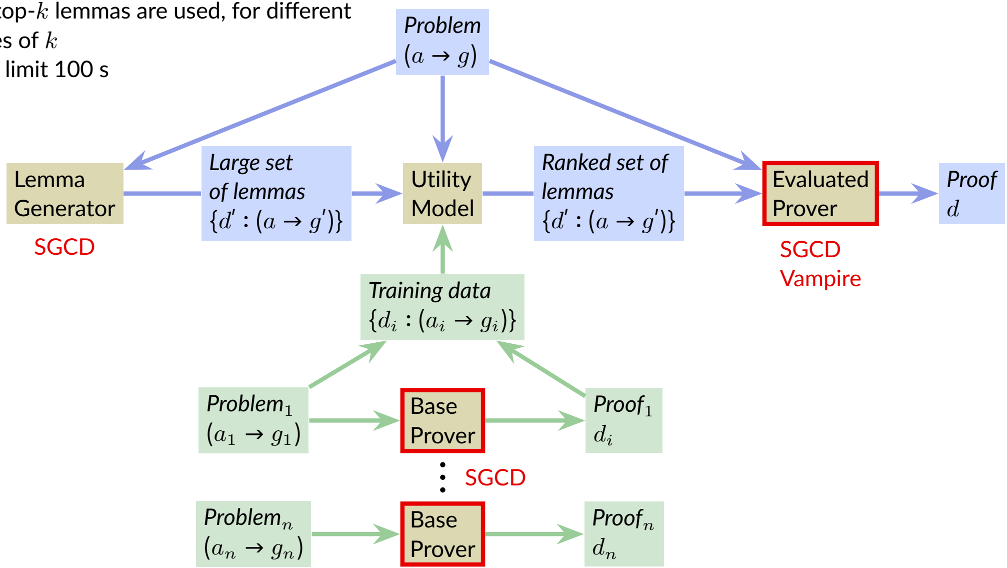


Experiment 2: Learned Lemmas to Enhance Other Provers



Experiment 3: Changing the Number of Added Lemmas

- The top-k lemmas are used, for different values of k
- Time limit 100 s



Experiment: Changing the Number of Added Lemmas

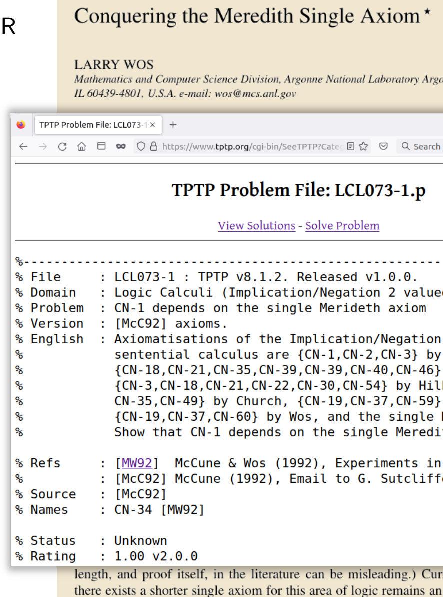
Prover	#Lemmas	#Solved problems
All problems		312
Vampire	Base	227
Vampire	10	226
Vampire	25	242
Vampire	50	246
Vampire	100	258
Vampire	200	257
Vampire	500	258
SGCD	Base	275
SGCD	10	278
SGCD	25	285
SGCD	50	284
SGCD	100	281
SGCD	200	283
SGCD	500	284

- **25 lemmas already yield substantial improvement**
- **Even 500 lemmas have no negative impact**

Proving LCL073-1

- Proven in ATP only by Wos in 2000 with several invocations of OTTER
- Proven **now with SGCD and replacing lemmas**
  - 98,198 lemmas generated by SGCD for **PSP-level**, cache limit 5,000, termination by exhaustion (60 s)
  - Ordered heuristically according to 5 general features (190 s)
  - The best 2,900 are supplied as replacing input lemmas to SGCD
  - SGCD called for proving: axiom-driven by **PSP-level**, goal-driven by **height**, cache limit 1,500, general heuristic restrictions (20 s)
  - **The structure of the proof reflects PSP-level plus one height step**

	Here	Wos	Meredith
Compacted size	**46**	74	40
Tree size	3,276	9,207	6,172
Height	40	48	30
Double negation	yes	no	yes
Max size of MGT of subproof	19	18	18


Conquering the Meredith Single Axiom *

LARRY WOS
Mathematics and Computer Science Division, Argonne National Laboratory Argo
IL 60439-4801, U.S.A. e-mail: wos@mcs.anl.gov

TPTP Problem File: LCL073-1.p

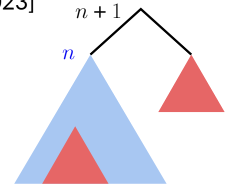
View Solutions - Solve Problem

```
%------------------------------------------------------------
% File     : LCL073-1 : TPTP v8.1.2. Released v1.0.0.
% Domain   : Logic Calculi (Implication/Negation 2 value
% Problem  : CN-1 depends on the single Merideth axiom
% Version  : [McC92] axioms.
% English  : Axiomatisations of the Implication/Negation
%            sentential calculus are {CN-1,CN-2,CN-3} by
%            {CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46}
%            {CN-3,CN-18,CN-21,CN-22,CN-30,CN-54} by Hil
%            CN-35,CN-49} by Church, {CN-19,CN-37,CN-59}
%            {CN-19,CN-37,CN-60} by Wos, and the single 
%            Show that CN-1 depends on the single Meredi

% Refs     : [MW92]  McCune & Wos (1992), Experiments in
%          : [McC92] McCune (1992), Email to G. Sutcliff
% Source   : [McC92]
% Names    : CN-34 [MW92]

% Status   : Unknown
% Rating   : 1.00 v2.0.0
```

length, and proof itself, in the literature can be misleading.) Cur
there exists a shorter single axiom for this area of logic remains an

The “Proof-Subproof” (PSP) Level Characterization – A Way of Inferencing Enabled by Proof Structure Terms

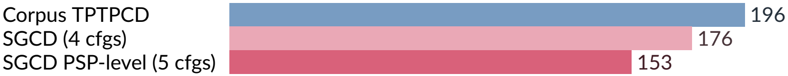
- A **principle observed** in proofs by Łukasiewicz and Meredith [CW,Bibel CADE 2021, 2023] **turned into a level characterization for SGCD**



D-terms in **PSP-level** $n + 1$ are those D-terms where

- one argument term is in PSP-level n
- and the other argument is a subterm of that term

- Enumeration by PSP-level
  - is incomplete (some D-terms are omitted)
  - has features of DAG enumeration: D-terms in PSP-level n have compacted size n
- Applications of enumeration by PSP-level
  - Solves ”Łukasiewicz’s single axiom” LCL038-1 with a short proof
  - Often applicable, often leads to proofs with small compacted size



  - Very useful for generating lemmas input to other provers
  - Key technique to solve ”Meredith’s single axiom” LCL073-1

Outlook, Follow-Up Steps

- **Learning from failure** [MR,CW,ZZ AITP 2023]
  - The residual of a failed proof attempt e.g. in SGCD consists of lemmas for the given axioms but other goals and can be used as training data
  - With the enhanced training data GNNs becomes superior to the linear models with handcrafted features
- Lemmas representing **proof compressions stronger than DAGs**
  - Nonunit lemmas corresponding to Horn clauses obtained with binary resolution upon the ternary *Detachment* clause
  - This may be handled via the connection structure calculus [Eder 1989] or via combinators in the D-terms [CW PAAR 2022]
  - It is not clear how important the stronger compressions are in practice
- **Beyond CD problems**
  - First-order Horn appears in close reach [CW PAAR 2022]
  - Witness Theory [Rezuş 2020] seems to consider theoretical generalizations of CD
  - Maybe also [Megill 1995]
  - Maybe the proof structures of the CM suffice
  - The axiom-driven mode of SGCD may be compatible with well-known techniques for equality handling

TPTP's CD Top

Problem	Rtg	C/T/H	Time	Prover
LCL426-1	1.00			
LCL425-1	1.00			
LCL421-1	1.00			
LCL420-1	1.00			
LCL419-1	1.00			
LCL418-1	1.00			
LCL073-1	1.00	46/3276/40	16.55	SGCD-HEU-3*
LCL063-1	1.00		943.481	E
LCL876+1	0.93	70/396/22	227.17	Prover9-HEU-1*
LCL422-1	0.86			
LCL417-1	0.86		647.386	Vampire-HEU-2*
LCL109-1	0.86	72/348/22	226.55	Prover9-HEU-1*
LCL428-1	0.57		0.227	E
LCL395-1	0.57	45/112/20	140.94	SGCD
LCL377-1	0.57	38/78/15	62.71	SGCD
LCL074-1	0.57	n 50/136/18	998.93	SGCD
LCL037-1	0.57	n 72/45359/39	172.29	Prover9
LCL875-1	0.43		0.298	Vampire
LCL394-1	0.43	41/81/17	267.22	SGCD
LCL376-1	0.43	30/76/15	58.17	SGCD-GNN*
LCL375-1	0.43	43/103/20	56.44	SGCD-LIN*
LCL374-1	0.43	33/77/17	42.47	SGCD

Problem	Rtg	C/T/H	Time	Prover
LCL167-1	0.43	48/265/22	27.53	SGCD-GNN*
LCL125-1	0.43	33/460/16	33.14	Prover9
LCL124-1	0.43	27/130/10	76.25	SGCD-LIN*
LCL062-1	0.43	44/115/21	285.10	SGCD-LIN*
LCL061-1	0.43	39/92/16	87.96	SGCD
LCL028-1	0.43	34/67/15	295.28	SGCD
LCL020-1	0.43	106/24989/37	21.65	Prover9-LIN*
LCL393-1	0.29	37/87/17	46.13	SGCD
LCL392-1	0.29	30/52/14	26.83	SGCD
LCL391-1	0.29	40/161/20	65.93	SGCD
LCL383-1	0.29	33/52/15	41.99	SGCD
LCL372-1	0.29	27/46/13	12.87	SGCD
LCL368-1	0.29	21/32/16	2.10	SGCD
LCL365-1	0.29	10/15/9	429.17	CCS-Vanilla
LCL119-1	0.29	83/28624/27	76.07	Prover9
LCL105-1	0.29	37/109/11	90.54	Prover9-LIN*
LCL099-1	0.29	20/41/6	459.30	SGCD
LCL032-1	0.29	n 67/15362/35	106.73	Prover9
LCL403-1	0.14	40/94/16	30.54	SGCD-LIN*
LCL390-1	0.14	31/45/14	281.13	SGCD
LCL384-1	0.14	13/23/5	683.01	CMProver
LCL382-1	0.14	29/53/18	6.21	SGCD

Summary of Contributions

- **Incorporation of proof structure terms into ATP with Machine Learning**
  - Consideration of features of proof structures
  - ATP/ML dataflow centered around the proof structure terms
- **Insights into the use of learned lemmas for provers of different paradigms and for different ways to incorporate lemmas**
  - SGCD is competitive with leading first-order provers
  - Learned lemmas improve Vampire substantially
  - The CM-CT provers without internal lemma maintenance are drastically improved, but still weak
  - Vampire and SGCD are able to handle a few hundreds of supplied lemmas
  - Linear and GNN models perform so far similarly
- **An ATP proof of LCL073-1**, a problem that was really hard for ATP
  - It is now solved by SGCD in a novel way that makes essential use of proof structure terms
- PS: everything is **implemented and freely available**

References I

[Bibel, 1987] Bibel, W. (1987).
Automated Theorem Proving.
Vieweg, Braunschweig.
First edition 1982.

[Bibel and Otten, 2020] Bibel, W. and Otten, J. (2020).
From Schütte's formal systems to modern automated deduction.
In Kahle, R. and Rathjen, M., editors, *The Legacy of Kurt Schütte*, chapter 13, pages 215–249. Springer.

[Eder, 1989] Eder, E. (1989).
A comparison of the resolution calculus and the connection method, and a new calculus generalizing both methods.
In Börger, E., Kleine Büning, H., and Richter, M. M., editors, *CSL '88*, volume 385 of *LNCS*, pages 80–98. Springer.

[Kaliszyk and Urban, 2015] Kaliszyk, C. and Urban, J. (2015).
Learning-assisted theorem proving with millions of lemmas.
J. Symb. Comput., 69:109–128.

[Letz, 1999] Letz, R. (1999).
Tableau and Connection Calculi. Structure, Complexity, Implementation.
Habilitationsschrift, TU München.
Available from `http://www2.tcs.ifi.lmu.de/~letz/habil.ps`, accessed Jun 30, 2022.

[Megill, 1995] Megill, N. D. (1995).
A finitely axiomatized formalization of predicate calculus with equality.
Notre Dame J. of Formal Logic, 36(3):435–453.

[Meredith and Prior, 1963] Meredith, C. A. and Prior, A. N. (1963).
Notes on the axiomatics of the propositional calculus.
Notre Dame J. of Formal Logic, 4(3):171–187.

[Rawson et al., 2023a] Rawson, M., Wernhard, C., and Zombori, Z. (2023a).
Learning to identify useful lemmas from failure.
In *AITP 2023 abstracts*.

[Rawson et al., 2023b] Rawson, M., Wernhard, C., Zombori, Z., and Bibel, W. (2023b).
Lemmas: Generation, selection, application.
CoRR, abs/2303.05854.
Submitted, preprint: `https://arxiv.org/abs/2303.05854`.

[Rezuş, 2020a] Rezuş, A. (2020a).
Tarski's Claim thirty years later (2010).
In [Rezuş, 2020b], pages 217–225.
Preprint (2016): `http://www.equivalences.org/editions/proof-theory/ar-tc-20160512.pdf`.

References III

[Rezuş, 2020b] Rezuş, A. (2020b).
Witness Theory – Notes on λ-calculus and Logic, volume 84 of *Studies in Logic*.
College Publications.

[Ulrich, 2001] Ulrich, D. (2001).
A legacy recalled and a tradition continued.
J. Autom. Reasoning, 27(2):97–122.

[Urban et al., 2008] Urban, J., Sutcliffe, G., Pudlák, P., and Vyskočil, J. (2008).
MaLARea SG1 – Machine Learner for Automated Reasoning with Semantic Guidance.
In Armando, A., Baumgartner, P., and Dowek, G., editors, *IJCAR 2008*, volume 5195 of *LNCS*, pages 441–456. Springer.

[Wernhard, 2022a] Wernhard, C. (2022a).
CD Tools – Condensed detachment and structure generating theorem proving (system description).
`https://arxiv.org/abs/2207.08453`.

[Wernhard, 2022b] Wernhard, C. (2022b).
Generating compressed combinatory proof structures – an approach to automated first-order theorem proving.
In Konev, B., Schon, C., and Steen, A., editors, *PAAR 2022*, volume 3201 of *CEUR Workshop Proc.* CEUR-WS.org.
Preprint: `https://arxiv.org/abs/2209.12592`.

References IV

[Wernhard, 2023] Wernhard, C. (2023).
Structure-generating first-order theorem proving.
In *AReCCa Workshop*.

[Wernhard and Bibel, 2021] Wernhard, C. and Bibel, W. (2021).
Learning from Łukasiewicz and Meredith: Investigations into proof structures.
In Platzer, A. and Sutcliffe, G., editors, *CADE 28*, volume 12699 of *LNCS (LNAI)*, pages 58–75. Springer.

[Wernhard and Bibel, 2023] Wernhard, C. and Bibel, W. (2023).
Investigations into proof structures.
Preprint, `http://cs.christophwernhard.com/papers/investigations/`.

[Wos, 2001] Wos, L. (2001).
Conquering the Meredith single axiom.
J. Autom. Reasoning, 27(2):175–199.