Structure-Generating Theorem Proving

Christoph Wernhard

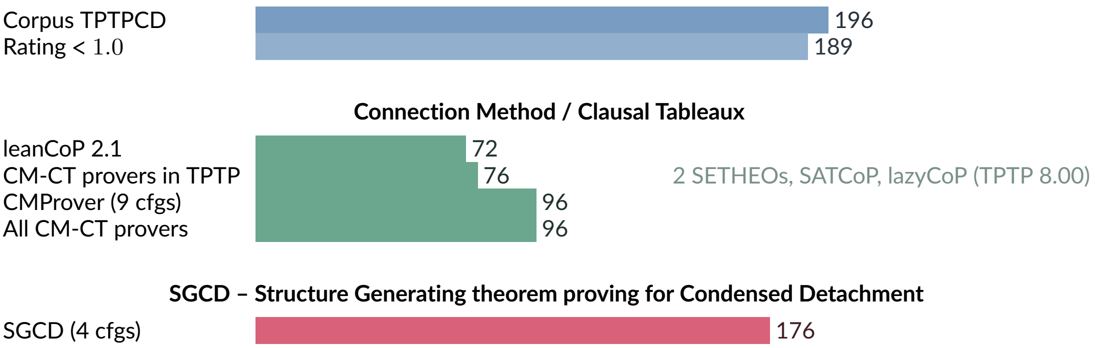
University of Potsdam

ARéCCa 2023

Prague, Czech Republic, Sep 18, 2023

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 457292495.
The work was supported by the North-German Supercomputing Alliance (HLRN).

Provers based on the connection method can be much much stronger than currently believed



Structure-Generating Theorem Proving

1. SGCD – Structure Generating Theorem Proving for Condensed Detachment
2. Experiments
3. Some Issues and Speculations
4. Conclusion

Background: CM-CT Provers

- Can be described as based on
  - Connection method (CM)
  - Clausal tableaux (CT)
  - Model elimination
- With systems such as
  - PTTP (Prolog Technology Theorem Prover) [Stickel 1988]
  - SETHEO [Letz, Bibel et al. 1992]
  - CMProver [CW 1992]
  - leanCoP [Otten, Bibel 2003]
  - nanoCoP, ileanCoP, MleanCoP, FEMaLeCoP, rlCoP, plCoP, lazyCoP, SATCoP, …
- **Enumerating proof trees** (instead of formulas like resolution)
  - Each proof structure appears there at most once
  - Interwoven with **unification of formulas** associated with nodes
  - Wrapped in **iterative deepening** upon size or height of the proof tree
- **Goal-driven**: top node initialized with Skolemized goal

Structure-Generating Proving: Core Ideas

- **Enumerating proof trees**, interwoven with **unification of formulas** associated with nodes

  Let's take this as starting point

- Wrapped in **iterative deepening** upon size or height of the proof tree

  This may be refined to grouping sets of structures into "levels"

- **Goal-driven**: top node initialized with Skolemized goal

  Let's combine this with axiom-driven operation, where proof-lemma pairs are enumerated

Let's start in a simplified setting where proof structures are available in a nice form

Formulas and Proof Structure Terms: Condensed Detachment (CD)

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

- Due to 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**
- 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
Goal	a negative ground unit

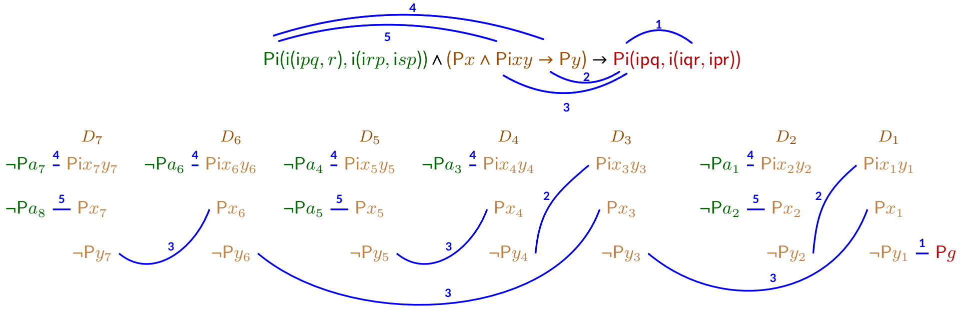
  Horn, first-order, binary function symbol, cyclic predicate dependency
- A possible inference system for CD

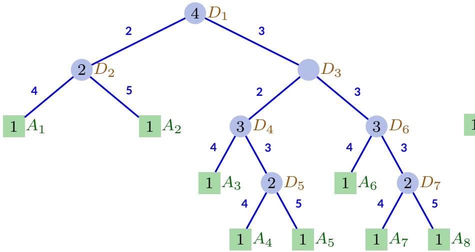
$$\frac{}{1 : \mathsf{P}(t)}\textit{fresh-copy} \quad \text{for the axiom } \mathsf{P}(t)$$

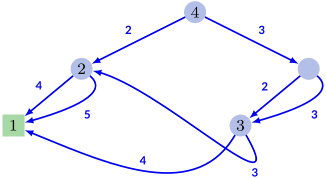
$$\frac{d_1 : \mathsf{P}(\mathsf{i}(x, y)) \qquad d_2 : \mathsf{P}(x')}{\mathsf{D}(d_1, d_2) : \mathsf{P}(y)}\,mgu(x, x')$$

- Hyperresolution also provides an inference system
- Relation to CM and more: [CW, Bibel CADE 21; 2023]

A Proof in Different Representations



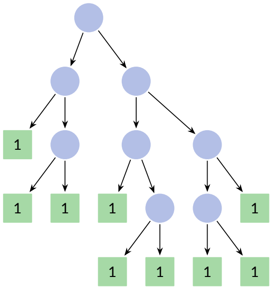


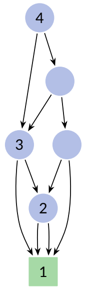


1. $CCCpqrCqr$
2. $CpCqp$ = D11
3. $CpCqCrp$ = D12

* 4. $CpCqCrCsCtCus$ = D2D33

Size Measures for D-Terms (Full Binary Trees)





Term representation

D(D(1, D(1, 1)), D(1, D(D(1, 1)), D(D(1, 1), 1)))

Representation by factor equations

$$
\begin{aligned}
2 &= \mathrm{D}(1, 1)\\
3 &= \mathrm{D}(1, 2)\\
4 &= \mathrm{D}(3, \mathrm{D}(3, \mathrm{D}(2, 1)))
\end{aligned}
$$

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

n		0	1	2	3	4	5	6
Tree size	OEIS:A000108	1	1	2	5	14	42	132
Height	OEIS:A001699	1	1	3	21	651	457,653	210,065,930,571
Compacted size	OEIS:A254789	1	1	3	15	111	1,119	14,487

Growth of the number of distinct D-terms for different size measures

Structure-Generating Proving: Core Ideas

- **Enumerating proof trees**, interwoven with **unification of formulas** associated with nodes

  Let's take this as starting point

- Wrapped in **iterative deepening** upon size or height of the proof tree

  This may be refined to grouping sets of structures into "levels"

- **Goal-driven**: top node initialized with Skolemized goal

  Let's combine this with axiom-driven operation, where proof-lemma pairs are enumerated

Let's start in a simplified setting where proof structures are available in a nice form

- Assume a Prolog predicate that **enumerates proof-MGT pairs for a given level**

$$\texttt{enum\_dterm\_mgt\_pairs}(+Level,\ ?DTerm,\ ?Formula)$$

- Level characterizations can be e.g. tree size or height of the D-term
- Depending on the parameter instantiation the predicate serves different purposes

$\texttt{enum\_dterm\_mgt\_pairs}(+Level,\ +Dterm,\ +Formula)$	verifying a proof
$\texttt{enum\_dterm\_mgt\_pairs}(+Level,\ +Dterm,\ -Formula)$	computing the MGT
$\texttt{enum\_dterm\_mgt\_pairs}(+Level,\ -Dterm,\ +Formula)$	**proving a formula (goal-driven)**
$\texttt{enum\_dterm\_mgt\_pairs}(+Level,\ -Dterm,\ -Formula)$	**generating lemmas (axiom-driven)**

- SGCD embeds it in **nested loops of goal- and axiom-driven phases**
- Its implementation can access a **cache** of solutions in lower levels
- The cache can be **heuristically restricted on the basis of MGTs**
- Optional: “lemma injection”: initializing the cache with given lemmas
- Optional: “hybrid levels”: different level characterizations for goal- and axiom-driven

$Cache := \varnothing$;
for $l := 0$ **to** $maxLevel$ **do**
  for $m := l$ **to** $l + preAddMaxLevel$ **do**
    $\texttt{enum\_dterm\_mgt\_pairs}(m, d, goal)$;
    throw $\texttt{proof\_found}(d)$
  $N := \{\langle l, d, f\rangle \mid \texttt{enum\_dterm\_mgt\_pairs}(l, d, f)\}$;
  if $N = \varnothing$ **then throw** $\texttt{exhausted}$;
  $Cache := \texttt{merge\_news\_into\_cache}(N, Cache)$

SGCD – Example of the Core Predicate for Maximal ("Up-To") Tree Size as Level Characterization

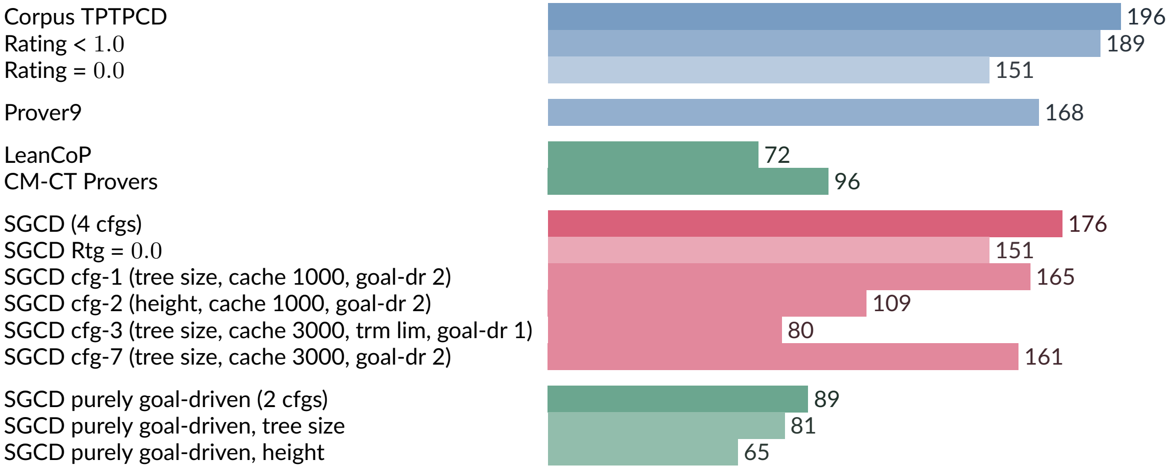
```
enum_dterm_mgt_pair(N, D, F) :-
        enum_dterm_mgt_pair_1(N, _, D, F).

enum_dterm_mgt_pair_1(N, N, I, F) :-
        id_axiom(I, F),              % Mapping of constant D-terms to axioms
        acyclic_term(F).             % Occurs check
enum_dterm_mgt_pair_1(N, N1, d(A,B), FY) :-
        N > 0,
        N2 is N - 1,
%       enum_dterm_mgt_pair_1(N2, N3, A, i(FX,FY)),
%       enum_dterm_mgt_pair_1(N3, N1, B, FX).
        pre_enum_dterm_mgt_pair_1(N2, N3, A, i(FX,FY)),
        pre_enum_dterm_mgt_pair_1(N3, N1, B, FX).

pre_enum_dterm_mgt_pair_1(N, N1, D, F) :-
        cached_level(N),
        !,
        level_solution(N2, F, D),
        N >= N2,
        acyclic_term(F),
        N1 is N-N2.
pre_enum_dterm_mgt_pair_1(N, N1, D, F) :-
        enum_dterm_mgt_pair_1(N, N1, D, F).
```

Implementation Aspects

- Implemented in **SWI-Prolog**
- Part of **CD Tools**, utilizes **PIE**
- CD Tools implements many concepts from [CW, Bibel 2023], which are available to SGCD e.g. for heuristic restrictions, e.g, regularity notions, variations of *organic*, n-simplification
- Free software `http://cs.christophwernhard.com/cdtools`
- Also tables and logs of experiments can be found on this website

SGCD on the 196 TPTPCD Problems

Corpus TPTPCD: Of the 206 **CD problems in the TPTP** exclude those 10 with: status *satisfiable*; detachment with disj. and neg.; goal theorem not an atom



SGCD for Theorem Finding

- **Łukasiewicz' 68 theses** were studied extensively by Wos with OTTER in the 1990s, they are now in the TPTP
- With a certain configuration of heuristics, **SGCD proves all of them in a single axiom-driven run in 2.5 min**
- OTTER solved all in a single run after the introduction of weight templates
- SGCD's proofs tend to larger compacted size but smaller tree size than those obtained by Wos in carefully crafted settings

SGCD Finds Small Proofs (1)

	C-avg	C-med	T-avg	T-med	H-avg	H-med
SGCD 4 config	15.83	13	29.36	17	8.52	6
Prover9	28.37	21	194,736.83	93	16.90	13
SGCD 4 config, after n-simplif.	15.80	12	29.36	17	8.52	6
Prover9, after n-simplif.	23.09	18	19,501.99	40	13.69	12

Proof sizes for the 163 problems provable by SGCD and Prover9

SGCD Finds Small Proofs (2)

- **CCS**, another prover in CD tools finds proofs with **guaranteed minimal compacted size** by enumeration upon compacted size

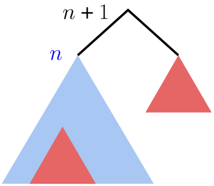
  This succeeds for 44% of the TPTPCD problems
- For many of the other problems **SGCD with PSP-level** as level characterization finds proofs with apparently small compacted size
- A particular example is **LCL038-1, for Łukasiewicz's single axiom** where a particular short proof is found

	C-size	T-size	Height
SGCD PSP-level	22	64	22
Meredith	31	491	29
Łukasiewicz	32	435	29
Prover9 after reductions by CD Tools	84	8,200	36
Prover9	93	216,094	40

Sizes of proofs of LCL038-1

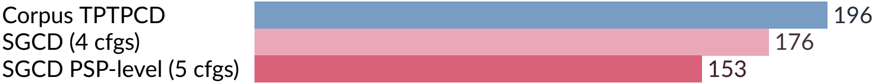
The “Proof-Subproof” (PSP) Level Characterization

- A principle **observed** in many steps of a proof by Łukasiewicz and a variation by Meredith [CW, Bibel CADE 2021] can be 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 leads to proofs with small compacted size
  - Generally often applicable



  - Very useful for generating lemmas input to other provers [Rawson, CW, Zombori, Bibel TABLEAUX 2023]
  - Key technique to solve ”Meredith’s single axiom” LCL073-1 [RWZB TABLEAUX 2023]

Issue: Stronger Proof Compressions

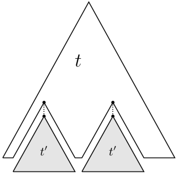


Fig. 3.1: A tree t containing two occurrences of the very same subtree t'.

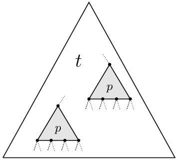


Fig. 3.2: A tree t containing two occurrences of the tree pattern p.

From [Lohrey et al. 2013] (on tree compression)

- So far we considered as lemmas **units** whose proof is a D-term;
  they are shared in the **unique minimal DAG representation** of the overall D-term
- In more general forms of lemmas **trees “with holes” are shared**
  - Horn clause lemmas (the body atoms correspond to the “holes”);
    obtained through binary resolution with the detachment clause
  - Tree grammars with variables (correspondoing to the “holes”) in nonterminals
  - **Combinators in D-terms** (“holes” are **mapped to subtree sharing in DAGs**) [CW PAAR 2022]
- The **connection structure calculus** [Eder 1989] considers such compressions; it can simulate resolution
- The **combinator approach** was implemented for first-order Horn [CW PAAR 2022]; it can simulate resolution

Issue: Stronger Proof Compressions – Some Open Questions

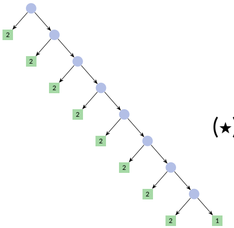
- The stronger compressions seem to require **enumeration by compacted size (DAG enumeration)**
  - Only a small DAG proof justifies the application of a compression (involvement of a combinator)
  - How to **combine goal- and axiom-driven modes for enumeration by compacted size**?
    For compacted size, a subproof can not be globally assigned a "level", but depending on context – if it already appeared in the proof under construction, it can be attached "for free"
- **How important are the strong compressions in practice?**

Goal: $\mathsf{P}(\mathsf{f}^8(\mathsf{a}))$ Axioms: $1\ \mathsf{P}(\mathsf{a})$

$2\ \mathsf{P}(x) \rightarrow \mathsf{P}(\mathsf{f}(x))$

A single structure proves Goal from Axioms:

$$2(2(2(2(2(2(2(21))))))) \qquad (\star)$$



Compound subterms are 21, $2(21)$, $2(2(21))$, $\ldots$ each occur once

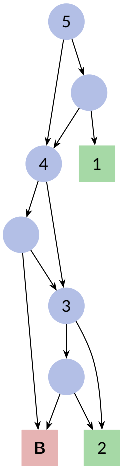
$$\mathbf{B} \stackrel{\text{def}}{=} \lambda xyz\,.\,x(yz)$$
$$\mathbf{B}xyz \rightarrow x(yz)$$

CL-term: Proof structure term in which **combinators** are permitted

$$\mathbf{B}(\mathbf{B}22)(\mathbf{B}22)(\mathbf{B}(\mathbf{B}22)(\mathbf{B}22)1) \qquad (\star\star)$$

- $(\star\star)$ **normalizes** with $\rightarrow$ to $(\star)$
- $(\star\star)$ has **multiple occurrences** of B … ultiple incoming edges in its minimal DAG
- In factors **B** has 2 arguments: **→ i**…
- $(\star\star)$ has compacted size 6, where … a)) with $n \geq 3$: CL-term with **B** has compacted si…
- For **determining the MGT** of a CL-… om identifiers, denoting their principal type

$$
\begin{array}{ll}
 & \mathbf{B}(\mathbf{B}22)(\mathbf{B}22)(\mathbf{B}(\mathbf{B}22)(\mathbf{B}22)1) \quad (\star\star)\\
\rightarrow & \mathbf{B}(\mathbf{B}22)(\mathbf{B}22)(\mathbf{B}22(\mathbf{B}221))\\
\rightarrow & \mathbf{B}(\mathbf{B}22)(\mathbf{B}22)(\mathbf{B}22(2(21)))\\
\rightarrow & \mathbf{B}(\mathbf{B}22)(\mathbf{B}22)(2(2(2(21))))\\
\rightarrow & \mathbf{B}22(\mathbf{B}22(2(2(2(21)))))\\
\rightarrow & \mathbf{B}22(2(2(2(2(2(21))))))\\
\rightarrow & 2(2(2(2(2(2(2(21))))))) \quad (\star)
\end{array}
$$

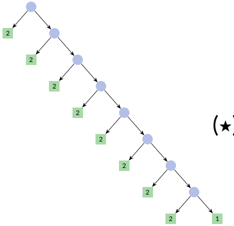


$3 = \mathbf{B}22$
$4 = \mathbf{B}33$
$5 = 4(41)$

Goal: $\mathsf{P}(\mathsf{f}^8(\mathsf{a}))$ Axioms: $1\ \mathsf{P}(\mathsf{a})$
$2\ \mathsf{P}(x) \rightarrow \mathsf{P}(\mathsf{f}(x))$

A single structure proves Goal from Axioms:

$$2(2(2(2(2(2(2(21))))))) \quad (\star)$$



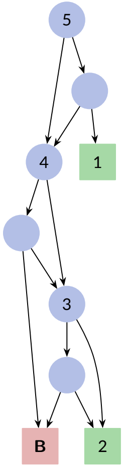
Compound subterms are 21, $2(21)$, $2(2(21))$, ... each occur once

$$\mathbf{B} \overset{\text{def}}{=} \lambda xyz\,.\,x(yz)$$
$$\mathbf{B}xyz \rightarrow x(yz)$$

CL-term: Proof structure term in which **combinators** are permitted

$$\mathbf{B}(\mathbf{B}22)(\mathbf{B}22)(\mathbf{B}(\mathbf{B}22)(\mathbf{B}22)1) \quad (\star\star)$$

- $(\star\star)$ **normalizes** with $\rightarrow$ to $(\star)$
- $(\star\star)$ has **multiple occurrences** of $\mathbf{B}22$ and $\mathbf{B}(\mathbf{B}22)(\mathbf{B}22)$, reflected in multiple incoming edges in its minimal DAG
- In factors **B** has 2 arguments: **$\rightarrow$ is not applicable *within* a factor**
- $(\star\star)$ has compacted size 6, where $(\star)$ has 8; generalizes to goals $\mathsf{P}(\mathsf{f}^{2^n}(\mathsf{a}))$ with $n \geq 3$: CL-term with **B** has compacted size $2n$
- For **determining the MGT** of a CL-term, combinators are taken like axiom identifiers, denoting their principal type



$3 = \mathbf{B}22$
$4 = \mathbf{B}33$
$5 = 4(41)$

Enumeration by Compacted Size: Blueprint for the Compiled Code

```
gen_d_mgt_upto_csize(N, D, F) :-
        gen_d_mgt_upto_csize_1(N, D,  _, [], _, F).

gen_d_mgt_upto_csize_1(N, I, N, L, L, F) :-
        axiom_id(F, I),
        acyclic_term(F).
gen_d_mgt_upto_csize_1(N, d(A,B), N1, L, [d(A,B)|L1], FY) :-
        N > 0,
        N0 is N-1,
        gen_d_mgt_upto_csize_2(N0, A, N2, L, L2, i(FX,FY)),
        gen_d_mgt_upto_csize_2(N2, B, N1, L2, L1, FX).

gen_d_mgt_upto_csize_2(N, D, N, L, L, F) :-
        member(D, L),
        d_mgt(D, F),
        acyclic_term(F).
gen_d_mgt_upto_csize_2(N, D, N1, L, L1, F) :-
        gen_d_mgt_upto_csize_1(N, D, N1, L, L1, F),
        not_abs_contains(L, D).
```

Improvements in the actual compilation results:

- No re-computation of the MGT of lemmas, just copying the lemmas
- Access to proper axioms and combinators is fully “unrolled” (not via `axiom_id/2`)
- Different predicates for each arity type

Issue: Systematization of Level Characterizations

n		0	1	2	3	4	5	6
Tree size	OEIS:A000108	1	1	2	5	14	42	132
Height	OEIS:A001699	1	1	3	21	651	457,653	210,065,930,571
Compacted size	OEIS:A254789	1	1	3	15	111	1,119	14,487
|PSP-level(n)|	OEIS:A001147	1	1	3	15	105	945	10,395

- **Disjoint vs cumulative** levels: e.g. tree size vs maximal tree size
- Interplay of levels with the **subterm relationship** (subterms required in a lower level?)
- **Gaps**: some intermediate level may have no member with MGT
- **Incompleteness**: e.g. PSP-level
- **"Context-dependency"**: why exactly is compacted size not suited for SGCD
- Is **PSP-level** a "context-independent" fragment of compacted size
- Can level characterizations be **combined**; beyond portfolio; beyond different ones for goal- and axiom-driven?
- Can **heuristic limitations** be considered in level characterizations?
- Relationship to **semi-naive evaluation**: delta-predicates keep preceding level for a triggering effect
- For a single problem we do not have decomposition into independent subproblems (subgoals may share variables) – can we get **decomposability when considering whole sets of problems (levels)** instead? (Computing a level via computing smaller levels that are independent from each other)

Issue: Generalization to Full First-Order Logic

- Bases
  - Witness theory [Rezu**s** 2020]
  - Finite axiomatization of predicate calculus [Megill 1995] (?)
  - CM, connection structure calculus
- CCS works for Horn
- Resolution proof translations
- Equality handling should be possible on the basis of the MGTs, like the heuristic restrictions

Structure-Generating Proving: Core Ideas

- **Enumerating proof trees**, interwoven with **unification of formulas** associated with nodes

  Let's take this as starting point

- Wrapped in **iterative deepening** upon size or height of the proof tree

  This may be refined to grouping sets of structures into "levels"

- **Goal-driven**: top node initialized with Skolemized goal

  Let's combine this with axiom-driven operation, where proof-lemma pairs are enumerated

Let's start in a simplified setting where proof structures are available in a nice form

References I

[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.

[Letz et al., 1992] Letz, R., Schumann, J., Bayerl, S., and Bibel, W. (1992).
SETHEO: A high-performance theorem prover.
J. Autom. Reasoning, 8(2):183–212.

[Lohrey et al., 2013] Lohrey, M., Maneth, S., and Mennicke, R. (2013).
XML tree structure compression using RePair.
Inf. Syst., 38(8):1150–1167.
System available from `https://github.com/dc0d32/TreeRePair`, 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.

References II

[Otten and Bibel, 2003] Otten, J. and Bibel, W. (2003).
leanCoP: lean connection-based theorem proving.
J. Symb. Comput., 36(1-2):139–161.

[Rawson et al., 2023] Rawson, M., Wernhard, C., Zombori, Z., and Bibel, W. (2023).
Lemmas: Generation, selection, application.
In Ramanayake, R. and Urban, J., editors, *TABLEAUX 2023*, LNAI.

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

[Schumann, 1994] Schumann, J. M. P. (1994).
DELTA – A bottom-up preprocessor for top-down theorem provers.
In *CADE-12*, volume 814 of *LNCS (LNAI)*, pages 774–777. Springer.

[Stickel, 1988] Stickel, M. E. (1988).
A Prolog technology theorem prover: implementation by an extended Prolog compiler.
J. Autom. Reasoning, 4(4):353–380.

References III

[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.

[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.
CoRR, abs/2304.12827.
Submitted, preprint: https://arxiv.org/abs/2304.12827.