Craig Interpolation and Query Reformulation with Clausal First-Order Tableaux

Christoph Wernhard, TU Dresden

TABLEAUX 2017, Brasília, 28 September 2017

1. Craig's Interpolation Theorem [Cra57]

Definition. Let F and G be first-order formulas such that $F \models G$. A **Craig-Lyndon interpolant** of F and G is a first-order formula H such that

1. $F \models H \models G$.
2. A predicate occurs positively (negatively) in H only if it occurs positively (negatively) in both F and G.
3. A function occurs in H only if it occurs in both F and G.

$$p \land q \models q \models q \lor r$$

2. Definiens Computation as Interpolation

The following statements are equivalent:

- $F \models \forall x\, (p(x) \leftrightarrow H)$, where p does not occur in H.
- H is an interpolant of $F \land p(x)$ and $\neg(F\{p \mapsto p'\} \land \neg p'(x))$.

Rationale: $F \land p(x) \models H$ iff $F \models p(x) \rightarrow H$ iff $F \models \forall x\, (p(x) \rightarrow H)$ ...

3. Construction of First-Order Craig Interpolants

- Basic approach: **extract of H from a proof of $F \models G$**
- Elegant tableau-based method in the textbooks [Smu68; Fit95]
  - Used as basis in works on query reformulation [TW11; Ben+16]
  - But hardly considered in automated reasoning, e.g., [BJ15; KV17]
- Principal limitation of resolution/superposition observed in [KV17]: There is no lower bound on the number of quantifier alterations in interpolants of a universal and an existential formula. Existing methods fail, e.g., on $\forall x\, p(r, x) \models \exists y \forall x\, p(y, x) \models \exists y\, p(y, b)$

Question: How to extract interpolants from clausal tableaux?

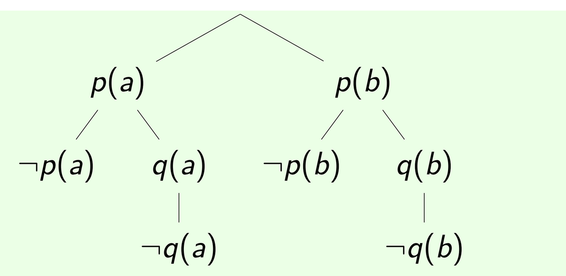
4. Clausal Tableaux [Let99a] (aka Clause Tableaux [Häh01])

Definition. A **clausal tableau for** a clausal formula F is a finite ordered tree whose nodes N with exception of the root are labeled with a first-order literal, denoted by $\mathrm{lit}(N)$, such that: For each node N the disjunction of the labels of all its children in their left-to-right order, denoted by $\mathrm{clause}(N)$, is an instance of a clause in F.

A node N is called **closed** if and only if it has an ancestor N' with $\mathrm{lit}(N') = \overline{\mathrm{lit}}(N)$. With a closed node N, a particular such ancestor N' is associated as the value of $\mathrm{target}(N)$. A tableau is called **closed** if and only if all of its leaves are closed.

The universal closure of a clausal formula F (with at least one constant) is **unsatisfiable** if and only if there exists a closed clausal tableau for F, if and only if there **exists a closed ground clausal tableau** for F, where the terms are formed from functions in F.

$$\begin{array}{ll} \forall x & ((p(a) \lor p(b)) \\ \land & (\neg p(x) \lor q(x)) \\ \land & \neg q(x)) \end{array} \models \bot$$



- Distinguishing features
  - from resolution: **clauses are not recombined**
  - from Smullyan/Fitting tableaux: Skolemization, only clausal formulas
- Many **efficient methods for automated theorem proving** can be viewed as constructing a clausal tableau
  - model elimination [Lov69], connection method [Bib81], Prolog technology theorem prover [Sti88] with systems like *PTTP* [Sti88] and *SETHEO* [Gol+94] and the living systems *leanCoP* [Ott10] with recent derivations [Kal15; KU15] and *CM* [DW97; Wer16]
  - bottom-up tableau construction SATCHMO [BM88], Hypertableaux [BFN96] with systems such as SATCHMO [BM88] and *E-KRHyper* [PW07]. (Today used for DL reasoners [MSH09])
  - many instance-based methods [BT10]

5. Interpolant Extraction from Clausal Ground Tableaux

Definition. A **colored clausal tableau for F_{red} and F_{blue}** is a clausal tableau for $F_{\mathrm{red}} \cup F_{\mathrm{blue}}$ whose nodes N, with exception of the root are labeled additionally with $\mathrm{color}(N) \in \{\mathrm{red}, \mathrm{blue}\}$ such that if N' is a child of N, then $\mathrm{clause}(N)$ is an instance of a clause in $F_{\mathrm{color}(N')}$.

Definition. $\mathrm{ipol}(N)$ is defined inductively for all nodes N of a closed colored clausal ground tableau:

i. If N is a leaf, then $\mathrm{ipol}(N)$ depends on $\mathrm{color}(N)$ and $\mathrm{color}(\mathrm{target}(N))$:

$\mathrm{color}(N)$	$\mathrm{color}(\mathrm{target}(N))$	$\mathrm{ipol}(N)$
red	red	$\bot$
red	blue	$\mathrm{lit}(N)$
blue	red	$\overline{\mathrm{lit}}(N)$
blue	blue	$\top$

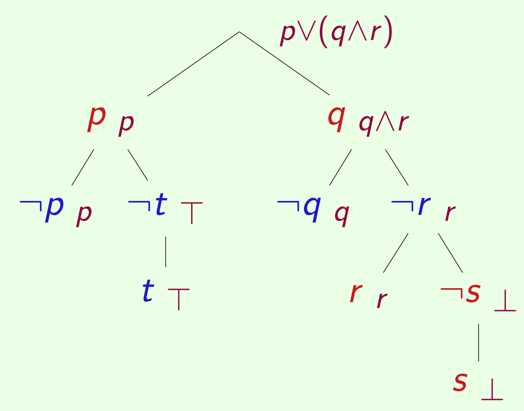
ii. If N is an inner node with children $N_1, \ldots, N_n$, then $\mathrm{ipol}(N)$ depends on the children's color:

$\mathrm{color}(N_1)$	$\mathrm{ipol}(N)$
red	$\bigvee_{i=1}^{n} \mathrm{ipol}(N_i)$
blue	$\bigwedge_{i=1}^{n} \mathrm{ipol}(N_i)$

Underlying Property. For all nodes N: $\mathrm{ipol}(N)$ is a Craig-Lyndon interpolant of $F_{\mathrm{red}} \land \mathrm{branch}_{\mathrm{red}}(N)$ and $\neg F_{\mathrm{blue}} \lor \mathrm{branch}_{\mathrm{blue}}(N)$, where $\mathrm{branch}_{Color}(N)$ is the conjunction of the literal labels of *Color* nodes on the branch to N (including N).

6. Example: Interpolant from Clausal Ground Tableaux

$$\begin{array}{ll} & (p \lor q) \land (\neg s \lor r) \land s \\ \models & p \lor (q \land r) \\ \models & (p \land t) \lor \neg t \lor (q \land r) \\ \equiv & \neg((\neg p \lor \neg t) \land t \land (\neg q \lor \neg r)) \end{array}$$



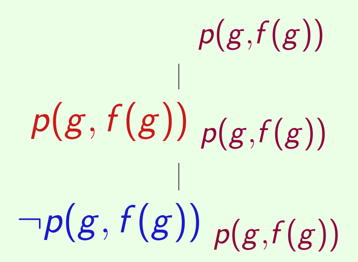
7. Lifting of Ground Interpolants

Conjecture. Let $F', \overline{G'}$ be clausal ground formulas obtained from first-order formulas $F, \neg G$ by Skolemization, clausification, copying clauses and instantiation. If H' is an interpolant of F' and G', then an interpolant H of F and G can be obtained as

$$Q_1 x_1 \ldots Q_n x_n H'\{t_n \mapsto x_n, \ldots, t_1 \mapsto x_1\},$$

where $t_1, \ldots, t_n$ are the outermost terms in H' whose principal functor h_i is a Skolem function, ordered such that if t_i is a subterm of t_j then $i < j$, and $Q_i = \exists$ ($Q_i = \forall$) if h_i was introduced with F' ($\overline{G'}$).

$$\begin{array}{ll} & \forall x \exists y\, p(x, y) \;\equiv\; \exists f \forall x\, p(x, f(x)) \\ \models & \forall x \exists y\, p(x, y) \\ \models & \forall x \exists y\, p(x, y) \equiv \neg \exists g \forall v\, \neg p(g, v) \\ & \forall x\, \exists y\, p(g, f(g))\{f(g) \mapsto x, g \mapsto y\} \end{array}$$



8. Access Interpolation [Ben+16]

Definition. **RQFO formulas** (FOL with Relativized Quantifiers) [Ben+16] are generated by

$$F \;::=\; \top \mid \bot \mid F \land F \mid F \lor F \mid \forall \boldsymbol{v}\, (\neg R \lor F) \mid \exists \boldsymbol{v}\, (R \land F),$$

where $\boldsymbol{v}$ is a (possibly empty) sequence of variables and relativizer R is a relational first-order atom in which all members of $\boldsymbol{v}$ do occur.

$$\forall x\, (\neg p(x) \lor \exists y\, (q(x, y) \land \top))$$

- The arguments of R not in $\boldsymbol{v}$ are "inputs", those in $\boldsymbol{v}$ "outputs"
- RQFO formulas can be **"evaluated"** with respect to finite relations

A **binding pattern** is a triple $\langle \mathit{Quantifier}, \mathit{Predicate}, \mathit{InputPositions} \rangle$.

The binding patterns of the example are $\langle \forall, p, \{\} \rangle$ and $\langle \exists, q, \{1\} \rangle$

A binding pattern is **covered** by another one with same quantifier and predicate and a subset of input positions.

$$\langle \exists, q, \{1, 2\} \rangle \text{ is covered by } \langle \exists, q, \{1\} \rangle$$

Definition. Let F and G be RQFO sentences such that $F \models G$. An **access interpolant** of F and G is a Craig-Lyndon H interpolant of F and G such that

1. H is an RQFO sentence.
2. Every universal (existential) binding pattern of H is covered by an universal (existential) binding pattern of F (G).

- Foundation of **query reformulation** [Ben+16]
- Definability is in the database context called *determinacy*
- Related: [Mar07; NSV10; Bor+10; TW11; BBC13; BCT14; HTW15]
- Generalizes Otto interpolation [Ott00]
- Constructive existence proof based on S/F tableaux [Ben+16]

Question: Can clausal tableaux be applied to access interpolation?

- Approach: simulating Smullyan/Fitting tableaux

9. Structure Preserving Clausification of RQFO Formulas

Definition (Sketch). For all subformula positions p of RQFO F:

- Let $D_p = \mathrm{d}_p(\mathbf{x}_p)$, where d_p is a fresh predicate and $\mathbf{x}_p$ are the free variables of $F|_p$.
- Define $\mathrm{def}_p^{\otimes}(F)$ for $\otimes \in \{\rightarrow, \leftarrow\}$ depending on the form of $F|_p$, e.g.,

$$\begin{array}{ll} \top & D_p \otimes \top \\ G \land H & \forall \mathbf{x}_p\, (D_p \otimes D_{p1} \land D_{p2}) \\ \exists x\, (R \land G) & \forall \mathbf{x}_p\, (D_p \otimes \exists x\, (R \land D_{p1})). \end{array}$$

Define

$$\begin{array}{lll} \mathrm{DEF}^{\rightarrow}(F) & \stackrel{\text{def}}{=} & \mathrm{d}_\epsilon \land \bigwedge_{p \in \mathrm{Positions}(F)} \mathrm{def}_p^{\rightarrow}(F). \\ \mathrm{DEF}^{\leftarrow}(F) & \stackrel{\text{def}}{=} & \neg \mathrm{d}_\epsilon \land \bigwedge_{p \in \mathrm{Positions}(F)} \mathrm{def}_p^{\leftarrow}(F). \end{array}$$

Let $\{p_0, \ldots, p_m\} = \mathrm{Positions}(F)$ and $\{q_0, \ldots, q_n\} = \mathrm{Positions}(G)$

$$\begin{array}{lrcl} & F \models & H & \models G \\ \textit{iff} & \exists \mathrm{d}_{p_1} \ldots \exists \mathrm{d}_{p_m} \mathrm{DEF}^{\rightarrow}(F) \models & H & \models \neg \exists \mathrm{e}_{q_1} \ldots \exists \mathrm{e}_{q_n} \mathrm{DEF}^{\leftarrow}(G) \\ \textit{iff} & \mathrm{DEF}^{\rightarrow}(F) \models & H & \models \neg \mathrm{DEF}^{\leftarrow}(G) \end{array}$$

Clausification of $\mathrm{DEF}^{\rightarrow}(F) \cup \mathrm{DEF}^{\leftarrow}(G)$ (with Skolemization applied to each conjunct individually) yields **clauses of certain forms** ($\mathcal{V}(F)$ is the set of variables in F, σ_p and γ_q map to Skolem terms):

1	D_ϵ		9	$\neg E_\epsilon$
2	$\neg D_p$		10	E_q
3	$\neg D_p \lor D_{p1}$	$\mathcal{V}(D_{p1}) \subseteq \mathcal{V}(D_p)$	11	$E_q \lor \neg E_{q1} \lor \neg E_{q2}$
4	$\neg D_p \lor D_{p2}$	$\mathcal{V}(D_{p2}) \subseteq \mathcal{V}(D_p)$	12	$E_q \lor \neg E_{q1}$
5	$\neg D_p \lor D_{p1} \lor D_{p2}$	$\mathcal{V}(D_{p1}) \cup \mathcal{V}(D_{p2}) \subseteq \mathcal{V}(D_p)$	13	$E_q \lor \neg E_{q2}$
6	$\neg D_p \lor \neg R_p \lor D_{p1}$	$\mathcal{V}(D_{p1}) \subseteq \mathcal{V}(D_p) \cup \mathcal{V}(R_p)$, $\mathcal{V}(R_p) \cup \mathcal{V}(D_{p1}) \subseteq \mathcal{V}(D_p) \cup \mathbf{v}_p$	14	$E_q \lor Q_q \gamma_q$
7	$\neg D_p \lor R_p \sigma_p$	$\mathcal{V}(R_p \sigma_p) \subseteq \mathcal{V}(D_p)$	15	$E_q \lor \neg E_{q1} \gamma_q$
8	$\neg D_p \lor D_{p1} \sigma_p$	$\mathcal{V}(D_{p1} \sigma_p) \subseteq \mathcal{V}(D_p)$	16	$E_q \lor \neg Q_q \lor \neg E_{q1}$

10. Interpolant Construction for RQFO Formulas

$\mathcal{T}(F)$ is the set of ground terms that occur as atom argument in F.

$$\mathcal{T}(\forall x\, p(x, f(y), a, g(b))) = \{a, g(b)\}$$

Definition. Let F and G be RQFO formulas such that $F \models G$. Let T be a colored ground tableau for two clausal formulas obtained by clausifying $\mathrm{DEF}^{\rightarrow}(F)$ and $\mathrm{DEF}^{\leftarrow}(G)$. Assume that T is

- closed, eager, regular
- leaf-only for the set of all $\neg D_p \rho$, $\neg R_p \rho$, $E_q \rho$, $\neg Q_q \rho$
- contiguous for the set of all pairs $\{R_p \sigma_p \rho, D_{p1} \sigma_p \rho\}$ and $\{Q_q \gamma_q \rho, \neg E_{p1} \gamma_q \rho\}$.

For inner nodes N of T with children $N_1, \ldots, N_k$ define $\text{acc-ipol}(N)$, depending on the form 1–16 of which $\mathrm{clause}(N)$ is an instance:

- **i.** Form 2–5, 7–8: $\text{acc-ipol}(N) \stackrel{\text{def}}{=} \bigvee_{i=2}^{k} \text{acc-ipol}(N_i)$.
- **ii.** Form 10–15: $\text{acc-ipol}(N) \stackrel{\text{def}}{=} \bigwedge_{i=2}^{k} \text{acc-ipol}(N_i)$.
- **iii.** Form 6,16: Let N' be the complementary ancestor of N_2.
  - a. If $\mathrm{color}(N') = \mathrm{color}(N_2)$, then $\text{acc-ipol}(N) \stackrel{\text{def}}{=} \text{acc-ipol}(N_3)$.
  - b. Form 6 with $\mathrm{clause}(N) = (\neg D_p \lor \neg R_p \lor D_{p1})\rho$, $\mathrm{color}(N_2) = \mathrm{red}$ and $\mathrm{color}(N') = \mathrm{blue}$:

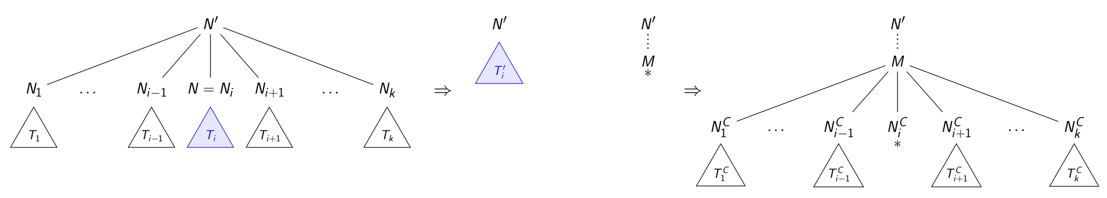
    $$\text{acc-ipol}(N) \stackrel{\text{def}}{=} \forall v_1 \ldots \forall v_n\, (\neg R_p \rho \lor \text{acc-ipol}(N_3))\Theta,$$

    where $\Theta = \{t_1 \mapsto v_1, \ldots, t_n \mapsto v_n\}$, $\{t_1, \ldots, t_n\} = \mathcal{T}(\neg R_p \rho) \setminus \mathcal{T}(\mathrm{DEF}^{\rightarrow}(F) \land \mathrm{branch}_{\mathrm{red}}(N))$ and $v_1, \ldots, v_n$ are fresh variables.
  - c. Form 16: analogously to 6, but with $\exists$

- Base cases are covered by forms 2 and 10 with $k = 1$
- Correctness proof with weakened notion of access interpolant that considers the "intermediate" forms of $\mathrm{DEF}^{\rightarrow}(F)$, $\mathrm{branch}_{\mathrm{red}}(N)$, $\mathrm{DEF}^{\leftarrow}(G)$ and $\mathrm{branch}_{\mathrm{blue}}(N)$

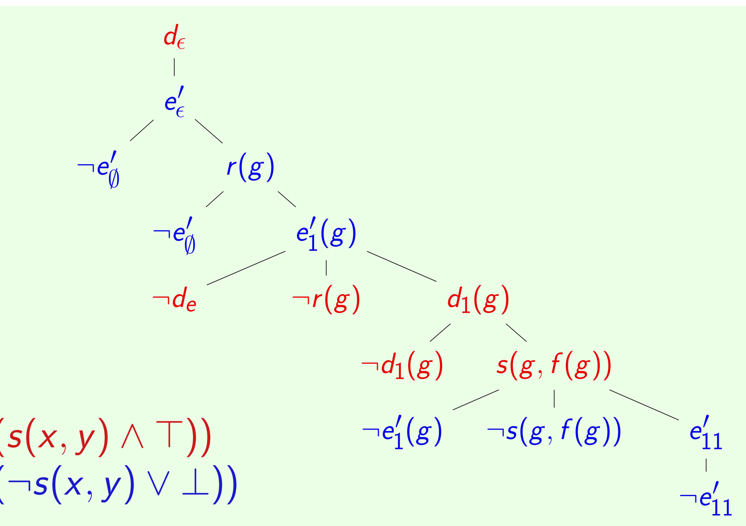
11. Tableau Properties and Conversions

- **Eager**: only leafs are closed (by simplification)
- **Regular**: no ancestor with same label (by simplification [Let99b])
- **Contiguous:** literals in S do not occur with an intermediate node on a branch (by simplification)
- **Leaf-only:** literals in S are only in leafs (expensive conversion)
  1. Let N be the inner node with literal is in S that is first visited in pre-order. Let N' be the parent of N
  2. Create a copy U of the subtree rooted at N'. In U remove the edges that originate in the node corresponding to N
  3. Replace the edges originating in N' with those originating in N
  4. For each leaf descendant M of N' that is complementary to N: attach a copy of U to M



12. Hypertableau Example

$$\begin{array}{ll} & \forall x\, (\neg r(x) \lor \exists y\, (s(x, y) \land \top)) \\ \models & \neg \exists x\, (r(x) \land \forall y\, (\neg s(x, y) \lor \bot)) \end{array}$$



13. Concluding Remarks

- Perspective different from calculi: tableau is given
- Re-combining clauses (resolution) at separate preprocessing of F and G
- Craig-Lyndon interpolation is implemented in *PIE* [Wer16]
- Planned: access interpolation with conversions and with hypertableaux

14. References

[BBC13] V. Bárány, M. Benedikt, and B. ten Cate. "Rewriting Guarded Negation Queries". In: *Mathematical Foundations of Computer Science 2013*. Vol. 8087. LNCS. Springer, 2013, pp. 98–110.

[BFN96] P. Baumgartner, U. Furbach, and I. Niemelä. "Hyper Tableaux". In: *JELIA'96*. Vol. 1126. LNCS (LNAI). Springer, 1996, pp. 1–17.

[BT10] P. Baumgartner and E. Thorstensen. "Instance Based Methods – A Brief Overview". In: *KI* 24.1 (Apr. 1, 2010), pp. 35–42.

[BCT14] M. Benedikt, B. ten Cate, and E. Tsamoura. "Generating low-cost plans from proofs". In: *PODS'14*. ACM, 2014, pp. 200–211.

[Ben+16] M. Benedikt et al. *Generating Plans from Proofs: The Interpolation-based Approach to Query Reformulation*. Morgan & Claypool, 2016.

[Bib81] W. Bibel. "On matrices with connections". In: *JACM* 28 (1981), pp. 633–645.

[BJ15] M. P. Bonacina and M. Johansson. "On Interpolation in Automated Theorem Proving". In: *J. Autom. Reasoning* 54.1 (2015), pp. 69–97.

[Bor+10] A. Borgida et al. "On Finding Query Rewritings under Expressive Constraints". In: *Proc. 18th Italian Symp. on Advanced Database Systems, SEBD 2010*. 2010.

[BM88] F. Bry and R. Manthey. "SATCHMO: A Theorem Prover Implemented in Prolog". In: *CADE-9*. Vol. 310. LNCS. Springer, 1988, pp. 415–434.

[Cra57] W. Craig. "Linear Reasoning. A New Form of the Herbrand-Gentzen Theorem". In: *J. Symb. Log.* 22.3 (1957), pp. 250–268.

[DW97] I. Dahn and C. Wernhard. "First Order Proof Problems Extracted from an Article in the Mizar Mathematical Library". In: *Int. Workshop on First-Order Theorem Proving, FTP'97*. RISC-Linz Report Series No. 97–50. Joh. Kepler Univ., Linz, Austria, 1997, pp. 58–62.

[Fit95] M. Fitting. *First-Order Logic and Automated Theorem Proving*. 2nd. Springer, 1995.

[Gol+94] C. Goller et al. "SETHEO V3.2: Recent Developments - System Abstract". In: *CADE-12*. Vol. 814. LNCS. Springer, 1994, pp. 778–782.

[Häh01] R. Hähnle. "Tableaux and Related Methods". In: *Handbook of Automated Reasoning*. Ed. by A. Robinson and A. Voronkov. Vol. 1. Elsevier, 2001. Chap. 3, pp. 101–178.

[HTW15] A. Hudek, D. Toman, and G. Wedell. "On Enumerating Query Plans Using Analytic Tableau". In: *TABLEAUX 2015*. Vol. 9323. LNCS (LNAI). Springer, 2015, pp. 339–354.

[Kal15] C. Kaliszyk. "Efficient Low-Level Connection Tableaux". In: *TABLEAUX 2015*. Vol. 9323. LNCS (LNAI). Springer, 2015, pp. 102–111.

[KU15] C. Kaliszyk and J. Urban. "FEMaLeCoP: Fairly Efficient Machine Learning Connection Prover". In: *LPAR-20*. Vol. 9450. LNCS. Springer, 2015, pp. 88–96.

[KV17] L. Kovács and A. Voronkov. "First-Order Interpolation and Interpolating Proofs Systems". In: *LPAR-21*. Vol. 26. EPiC. 2017, pp. 49–64.

[Let99a] R. Letz. "First-Order Tableau Methods". In: *Handbook of Tableau Methods*. Ed. by R. H. Marcello D'Agostino Dov M. Gabbay and J. Posegga. Kluwer Academic Publishers, 1999, pp. 125–196.

[Let99b] R. Letz. "Tableau and Connection Calculi. Structure, Complexity, Implementation". Habilitationsschrift. TU München, 1999.

[Lov69] D. W. Loveland. "A Simplified Format for the Model Elimination Theorem-Proving Procedure". In: *JACM* 16.3 (1969), pp. 349–363.

[Mar07] M. Marx. "Queries determined by views: Pack your views". In: *PODS '07*. ACM, 2007, pp. 23–30.

[MSH09] B. Motik, R. Shearer, and I. Horrocks. "Hypertableau Reasoning for Description Logics". In: *JAIR* 36 (2009), pp. 165–228.

[NSV10] A. Nash, L. Segoufin, and V. Vianu. "Views and queries: Determinacy and rewriting". In: *ACM Transactions on Database Systems* 35.3 (2010).

[Ott10] J. Otten. "Restricting backtracking in connection calculi". In: *AI Communications* 23.2-3 (2010), pp. 159–182.

[Ott00] M. Otto. "An Interpolation Theorem". In: *Bulletin of Symbolic Logic* 6 (4 2000), pp. 447–462.

[PW07] B. Pelzer and C. Wernhard. "System Description: E-KRHyper". In: *CADE-21*. Vol. 4603. LNCS (LNAI). Springer, 2007, pp. 503–513.

[Smu68] R. M. Smullyan. *First-Order Logic*. Also republished with corrections by Dover publications, New York, 1995. New York: Springer, 1968.

[Sti88] M. E. Stickel. "A Prolog technology theorem prover: implementation by an extended Prolog compiler". In: *J. Autom. Reasoning* 4.4 (1988), pp. 353–380.

[TW11] D. Toman and G. Weddell. *Fundamentals of Physical Design and Query Compilation*. Morgan and Claypool, 2011.

[Wer16] C. Wernhard. "The PIE system for Proving, Interpolating and Eliminating". In: *PAAR-2016*. CEUR Workshop Proceedings 1635. Aachen, 2016, pp. 125–138.

Acknowledgment: This work was supported by DFG grant WE 5641/1-1