7.17 SCLT Clause Learning from Simple Models Modulo Theories

Let $\mathcal{T}^{\mathcal{B}}$ be first-order logic *background theory* over signature $\Sigma^{\mathcal{B}} = (\mathcal{S}^{\mathcal{B}}, \Omega^{\mathcal{B}}, \Pi^{\mathcal{B}})$ and term-generated $\Sigma^{\mathcal{B}}$-algebras $\mathcal{C}^{\mathcal{B}}$: $\mathcal{T}^{\mathcal{B}} = (\Sigma^{\mathcal{B}}, \mathcal{C}^{\mathcal{B}})$. A constant $c \in \Omega^{\mathcal{B}}$ is called a *domain constant* if $c^{\mathcal{A}} \neq d^{\mathcal{A}}$ for all $\mathcal{A} \in \mathcal{C}^{\mathcal{B}}$ and for all $d \in \Omega^{\mathcal{B}}$ with $d \neq c$. Let $\Sigma^{\mathcal{F}} = (\mathcal{S}^{\mathcal{F}}, \Omega^{\mathcal{F}}, \Pi^{\mathcal{F}})$ be a *foreground signature* with respect to $\mathcal{T}^{\mathcal{B}}$ where $\mathcal{S}^{\mathcal{B}} \subseteq \mathcal{S}^{\mathcal{F}}$, $\Omega^{\mathcal{B}} \cap \Omega^{\mathcal{F}} = \emptyset$, and $\Pi^{\mathcal{B}} \cap \Pi^{\mathcal{F}} = \emptyset$.

Definition 7.17.1 (Hierarchic Specification)**.** A *hierarchic specification* is a pair $\mathcal{H} = (\mathcal{T}^{\mathcal{B}}, \Sigma^{\mathcal{F}})$ with associated signature $\Sigma^{\mathcal{H}} = (\mathcal{S}^{\mathcal{F}}, \Omega^{\mathcal{B}} \cup \Omega^{\mathcal{F}}, \Pi^{\mathcal{B}} \cup \Pi^{\mathcal{F}})$. It generates *hierarchic* $\Sigma^{\mathcal{H}}$-algebras. A $\Sigma^{\mathcal{H}}$-algebra $\mathcal{A}$ is called *hierarchic* with respect to its background theory $\mathcal{T}^{\mathcal{B}}$, if $\mathcal{A}^{\mathcal{H}}|_{\Sigma^{\mathcal{B}}} \in \mathcal{C}^{\mathcal{B}}$.

As usual, $\mathcal{A}^{\mathcal{H}}|_{\Sigma^{\mathcal{B}}}$ is obtained from a $\mathcal{A}^{\mathcal{H}}$-algebra by removing all carrier sets $S^{\mathcal{A}}$ for all $S \in (\mathcal{S}^{\mathcal{F}} \setminus \mathcal{S}^{\mathcal{B}})$, all functions from $\Omega^{\mathcal{F}}$ and all predicates from $\Pi^{\mathcal{F}}$. We write $\models_{\mathcal{H}}$ for the entailment relation with respect to hierarchic algebras and formulas from $\Sigma^{\mathcal{H}}$ and $\models_{\mathcal{B}}$ for the entailment relation with respect to the $\mathcal{C}^{\mathcal{B}}$ algebras and formulas from $\Sigma^{\mathcal{B}}$.

Terms, atoms, literals build over $\Sigma^{\mathcal{B}}$ are called *pure background terms*, *pure background atoms*, and *pure background literals*, respectively. All terms, atoms, with a top-symbol from $\Omega^{\mathcal{B}}$ or $\Pi^{\mathcal{B}}$, respectively, are called *background terms*, *background atoms*, respectively. A background atom or its negation is a *background literal*. All terms, atoms, with a top-symbol from $\Omega^{\mathcal{F}}$ or $\Pi^{\mathcal{F}}$, respectively, are called *foreground terms*, *foreground atoms*, respectively. A foreground atom or its negation is a *foreground literal*. Given a set (sequence) of $\mathcal{H}$ literals, the function bgd returns the set (sequence) of background literals and the function fgd the respective set (sequence) of foreground literals.

As a running example, I consider in detail the Bernays-Schoenfinkel clause fragment over linear arithmetic: BS(LRA). The background theory is linear rational arithmetic over the many-sorted signature $\Sigma^{\mathrm{LRA}} = (\mathcal{S}^{\mathrm{LRA}}, \Omega^{\mathrm{LRA}}, \Pi^{\mathrm{LRA}})$ with $\mathcal{S}^{\mathrm{LRA}} = \{\mathrm{LRA}\}$, $\Omega^{\mathrm{LRA}} = \{0, 1, +, -\} \cup \mathbb{Q}$, $\Pi^{\mathrm{LRA}} = \{\leq, <, \neq, =, >, \geq\}$) where LRA is the linear arithmetic sort, the function symbols consist of $0, 1, +, -$ plus the rational numbers and predicate symbols $\leq, <, =, \neq, >, \geq$. The linear arithmetic theory $\mathcal{T}^{\mathrm{LRA}} = (\Sigma^{\mathrm{LRA}}, \{\mathcal{A}^{\mathrm{LRA}}\})$ consists of the linear arithmetic signature together with the standard model $\mathcal{A}^{\mathrm{LRA}}$ of linear arithmetic. This theory is then extended by the free (foreground) first-order signature $\Sigma^{\mathrm{BS}} = (\{\mathrm{LRA}\}, \Omega^{\mathrm{BS}}, \Pi^{\mathrm{BS}})$ where Ω^{BS} is a set of constants of sort LRA different from Ω^{LRA} constants, and Π^{BS} is a set of first-order predicates over the

sort LRA. We are interested in hierarchic algebras $\mathcal{A}^{\mathrm{BS(LRA)}}$ over the signature $\Sigma^{\mathrm{BS(LRA)}} = (\{\mathrm{LRA}\}, \Omega^{\mathrm{BS}} \cup \Omega^{\mathrm{LRA}}, \Pi^{\mathrm{BS}} \cup \Pi^{\mathrm{LRA}})$ that are $\Sigma^{\mathrm{BS(LRA)}}$ algebras such that $\mathcal{A}^{\mathrm{BS(LRA)}}|_{\Sigma^{\mathrm{LRA}}} = \mathcal{A}^{\mathrm{LRA}}$.

Definition 7.17.2 (Simple Substitutions)**.** A substitution σ is called *simple* if $x_S\sigma \in T_S(\Sigma^{\mathcal{B}}, \mathcal{X})$ for all $x_S \in \mathrm{dom}(\sigma)$ and $S \in \mathcal{S}^{\mathcal{B}}$.

As usual, clauses are disjunctions of literals with implicitly universally quantified variables. We often write a $\Sigma^{\mathcal{H}}$ clause as a *constrained clause*, denoted $\Lambda \parallel C$ where Λ is a conjunction of background literals and C is a disjunction of foreground literals semantically denoting the clause $\neg\Lambda \vee C$. A *constrained closure* is denoted as $\Lambda \parallel C \cdot \sigma$ where σ is grounding for Λ and C. A constrained closure $\Lambda \parallel C \cdot \sigma$ denotes the ground constrained clause $\Lambda\sigma \parallel C\sigma$.

In addition, we assume a well-founded, total, strict ordering $\prec$ on ground literals, called an $\mathcal{H}$-order, such that background literals are smaller than foreground literals. This ordering is then lifted to constrained clauses and sets thereof by its respective multiset extension. We overload $\prec$ for literals, constrained clauses, and sets of constrained clause if the meaning is clear from the context. We define $\preceq$ as the reflexive closure of $\prec$ and $N^{\preceq \Lambda \parallel C} := \{D \mid D \in N \text{ and } D \preceq \Lambda \parallel C\}$. For example, an instance of an LPO with according precedence can serve as $\prec$.

Definition 7.17.3 (Abstracted/Pure Clause)**.** A clause $\Lambda \parallel C$ is *abstracted* if the arguments of $\mathcal{S}^{\mathcal{B}}$ sort of any predicate from $\Pi^{\mathcal{F}}$ in an atom in C are only variables. $\Lambda \parallel C$ is called *pure* if it does not contain symbols from $\Omega^{\mathcal{F}}$ ranging into a sort of $\mathcal{S}^{\mathcal{B}}$.

These two notions are extended to clause sets in the natural way. Any clause set can be transformed into an abstracted clause set.

Abstraction $N \uplus \{C \vee E[t]_p[s]_q\} \Rightarrow_{\mathrm{ABSTR}} N \cup \{C \vee x_s \not\approx s \vee E[x_S]_q\}$ provided t, s are non-variable terms, $q \not< p$, $\mathrm{sort}(s) = S$, and either $\mathrm{top}(t) \in \Sigma^F$ and $\mathrm{top}(s) \in \Sigma^B$ or $\mathrm{top}(t) \in \Sigma^B$ and $\mathrm{top}(s) \in \Sigma^F$

In case of BS(LRA) abstraction can only be applied to constants below a predicate.

Definition 7.17.4 (Clause Redundancy)**.** A ground constrained clause $\Lambda \parallel C$ is *redundant* with respect to a set N of ground constrained clauses and an order $\prec$ if $N^{\preceq \Lambda \parallel C} \models_{\mathcal{H}} \Lambda \parallel C$. A clause $\Lambda \parallel C$ is *redundant* with respect to a clause set N, an $\mathcal{H}$-order $\prec$, and a set of constants B if for all $\Lambda' \parallel C' \in \mathrm{grd}((\mathcal{S}^{\mathcal{F}}, B, \Pi^{\mathcal{B}} \cup \Pi^{\mathcal{F}}), \Lambda \parallel C)$ the clause $\Lambda' \parallel C'$ is redundant with respect to $\cup_{D \in N} \mathrm{grd}((\mathcal{S}^{\mathcal{F}}, B, \Pi^{\mathcal{B}} \cup \Pi^{\mathcal{F}}), D)$.

Assumption 7.17.5 (Considered Clause Sets)**.** For the rest of this section I consider only pure, abstracted clause sets N. I assume that the background theory $\mathcal{T}^{\mathcal{B}}$ is term-generated, compact, contains an equality $=$, and that all constants of the background signature are domain constants. I further assume that the set $\Omega^{\mathcal{F}}$ contains infinitely many constants for each background sort.

Example 7.17.6 (Pure Clauses)**.** With respect to BS(LRA) the unit clause $x \geq 5, 3x + 4y = z \parallel Q(x, y, z)$ is abstracted and pure while the clause $x \geq 5, 3x + 4y = a, z = a \parallel Q(x, y, z)$ is abstracted but not pure because of the foreground constant a of the LRA sort, and the clause $x \geq 5, 3x + 4y = 7 \parallel Q(x, y, 7)$ is not abstracted.

Note that for pure, abstracted clause sets, any unifier between two foreground literals is simple and its codomain consists of variables only.

In order for the SCL(T) calculus to be effective, decidability in $\mathcal{T}^{\mathcal{B}}$ is needed as well. For the calculus we implicitly use the following equivalence: A $\Sigma^{\mathcal{B}}$ sentence

$$\exists x_1, \ldots, x_n \phi$$

where ϕ is quantifier free is true, i.e., $\models_{\mathcal{B}} \exists x_1, \ldots, x_n \phi$ iff the ground formula

$$\phi\{x_1 \mapsto a_1, \ldots, x_n \mapsto a_n\}$$

where the a_i are $\Omega^{\mathcal{F}}$ constants of the respective background sorts is $\mathcal{H}$ satisfiable. Together with decidability in $\mathcal{T}^{\mathcal{B}}$ this guarantees decidability of the satisfiability of ground constraints from constrained clauses.

If not stated otherwise, satisfiability means satisfiability with respect to $\mathcal{H}$. The function adiff(B) for some finite sequence of background sort constants denotes a constraint that implies different interpretations for the constants in B. In case the background theory enables a strict ordering $<$ as LRA does, then the ordering can be used for this purpose. For example, adiff$([a, b, c])$ is then the constraint $a < b < c$. In case the background theory does not enable a strict ordering, then inequations can express disjointness of the constants. For example, adiff$([a, b, c])$ is then constraint $a \neq b \wedge a \neq c \wedge b \neq c$. An ordering constraint has the advantage over an inequality constraint that it also breaks symmetries. Assuming all constants to be different will eventually enable a satisfiability test for foreground literals based on purely syntactic complementarity.

The inference rules of SCL(T) are represented by an abstract rewrite system. They operate on a problem state, a six-tuple $\Gamma = (M; N; U; B; k; D)$ where M is a sequence of annotated ground literals, the *trail*; N and U are the sets of *initial* and *learned* constrained clauses; B is a finite sequence of constants of background sorts for instantiation; k counts the number of decisions in M; and D is a constrained closure that is either $\top$, $\Lambda \parallel \bot \cdot \sigma$, or $\Lambda \parallel C \cdot \sigma$. Foreground literals in M are either annotated with a number, a level; i.e., they have the form L^k meaning that L is the k-th guessed decision literal, or they are annotated with a constrained closure that propagated the literal to become true, i.e., they have the form $(L\sigma)^{(\Lambda \parallel C \vee L) \cdot \sigma}$. An annotated literal is called a decision literal if it is of the form L^k and a propagation literal or a propagated literal if it of in the form $L \cdot \sigma^{(\Lambda \parallel C \vee L) \cdot \sigma}$. A ground foreground literal L is of *level* i with respect to a problem state $(M; N; U; B; k; D)$ if L or comp(L) occurs in M and the first decision literal left from L (comp(L)) in M, including L, is annotated with i. If there is no such decision literal then its level is zero. A ground constrained clause $\Lambda \parallel C$ is of *level* i with respect to a problem state $(M; N; U; B; k; D)$ if

i is the maximal level of a foreground literal in C; the level of an empty clause $\Lambda \parallel \bot \cdot \sigma$ is 0. A ground literal L is *undefined* in M if neither L nor $\mathrm{comp}(L)$ occur in M. The initial state for a first-order, pure, abstracted $\mathcal{H}$ clause set N is $(\epsilon; N; \emptyset; B; 0; \top)$, where B is a finite sequence of foreground constants of background sorts. These constants cannot occur in N, because N is pure. The final state $(\epsilon; N; U; B; 0; \Lambda \parallel \bot)$ denotes unsatisfiability of N. Given a trail M and its foreground literals $\mathrm{fgd}(M) = \{L_1, \ldots, L_n\}$ an $\mathcal{H}$ ordering $\prec$ *induced* by M is any $\mathcal{H}$ ordering where $L_i \prec L_j$ if L_i occurs left from L_j in M, or, L_i is defined in M and L is not.