Saturation up to Redundancy for Tableau and Sequent Calculi

Martin Giese

Johann Radon Institute for Computational and Applied Mathematics
Altenbergerstr. 69, A-4040 Linz, Austria
martin.giese@oeaw.ac.at

Abstract. We discuss an adaptation of the technique of saturation up to redundancy, as introduced by Bachmair and Ganzinger [1], to tableau and sequent calculi for classical first-order logic. This technique can be used to easily show the completeness of optimized calculi that contain destructive rules e.g. for simplification, rewriting with equalities, etc., which is not easily done with a standard Hintikka-style completeness proof. The notions are first introduced for Smullyan-style ground tableaux, and then extended to constrained formula free-variable tableaux.

1 Introduction

The usual Hintikka-style completeness proof for tableau or sequent calculi requires branches to be *saturated*. This means that for any formula appearing on a branch and any inference possible on that formula, all formulae introduced by that inference on at least one of the created branches also appear on the branch.

While this condition poses no problem in the standard calculi for classical logic, more complicated calculi might allow several different inferences on the same formula. In that case, none of these inferences may in general delete the original formula, since it has to remain available for the other inferences to achieve saturation.

In many cases, destructive rules would make a calculus more efficient. Examples are rewriting with equalities [5], type reasoning [7], as well as various domain specific calculi, see e.g. [3], which all use non-destructive rules. The completeness of destructive variants of these calculi cannot be shown using a Hintikka-style proof. Sometimes, proof transformation techniques can be used to cope with destructiveness, see e.g. [6], but these require plenty of creativity and are very specific to the calculus at hand.

In the context of resolution theorem proving, Bachmair and Ganzinger have established the admirable framework of *saturation up to redundancy*, see e.g. [1]. The idea is that a clause can be deleted from a clause set if it is *redundant* with respect to the other clauses. Precise definitions are given for what constitutes a valid redundancy criterion, and then completeness is shown for all inference systems that obey certain restrictions.

In this paper, the results of Bachmair and Ganzinger are transferred to the setting of tableau and sequent calculi for classical first-order logic. After introducing in Sect. 2 some basic notions about the type of calculi we are going to consider, Sect. 3 presents notions of redundancy and a generic completeness theorem for tableaux. In the next two sections, the technique is demonstrated on two simple calculi. We then extend our notions to free variable calculi in Sect. 6. This is again followed by a case study, before we conclude the paper in Sect. 8.

2 Semi-sequent calculi

We simplify our presentation by considering only *semi-sequent* calculi. A semi-sequent calculus is like a sequent calculus in which the right hand side, the succedent, of every sequent is empty. Such calculi are also known as *block tableau* [8] calculi.

Definition 1. *1 A* semi-sequent *is a set of formulae written* $\phi_1, \ldots, \phi_n \vdash$. *With the notation* $\phi_1, \ldots, \phi_n, \Gamma \vdash$, *we mean a semi-sequent that consists of the set of formulae* $\{\phi_1, \ldots, \phi_n\} \cup \Gamma$.

Definition 2. *A* tableau *for a semi-sequent calculus is a tree where each node is labeled with a semi-sequent.*

A derivation *consists of a sequence of tableaux, each of which is constructed from the previous one through the application of an inference on one of the leaves. The first tableau consists of only the root node, which is labeled with the* initial semi-sequent. *A* derivation for a formula ϕ *is a derivation with initial semi-sequent* $\phi \vdash$.

A semi-sequent is called closed *if it contains* $\bot$, *the false formula, otherwise it is called* open. *A tableau is called closed, if the semi-sequents in all leaves are closed.*

The general form of an inference in a semi-sequent calculus is

$$\frac{\phi_{11}, \ldots, \phi_{1m_1}, \Gamma \vdash \quad \cdots \quad \phi_{n1}, \ldots, \phi_{nm_n}, \Gamma \vdash}{\phi_{01}, \ldots, \phi_{0m_0}, \Gamma \vdash}$$

We refer to the upper semi-sequents as *premises* and the lower one as *conclusion* of the inference. One of the formulae in the conclusion of every inference is identified and called the *main formula* of the conclusion, the others are the *side formulae*.

Application of such an inference requires all formulae $\phi_{01}, \ldots, \phi_{0m_0}$ of the conclusion to be present in a leaf semi-sequent. The tree is then expanded by appending n children to the leaf containing the modified sequents given by the premises.[1]

[1] It may be a bit confusing that proof construction starts from the conclusion and adds more premises, but this is the common terminology in sequent calculi. A possible reading is "to conclude that Γ_0 is unsatisfiable, we have to show that Γ_1 to Γ_n are unsatisfiable."

Given a finite or infinite derivation $(\mathcal{T}_i)_{i\in\mathbb{N}}$, we can easily define its limit $\mathcal{T}^\infty$, which may in general be an infinite tree. This possibly infinite tree consists of possibly infinitely many possibly infinite branches, which again are sequences $(\Gamma_i)_{i\in\mathbb{N}}$ of semi-sequents. While each tableau in the derivation is contained in each of its successors, this is not necessarily the case for the semi-sequents on a tableau branch: inferences might remove formulae from semi-sequents. Still, one can form a limit semi-sequent,

$$\Gamma^\infty := \bigcup_{i\in\mathbb{N}} \bigcap_{j\geq i} \Gamma_j$$

consisting of all *persistent* formulae on the branch.

3 Redundancy

The definitions, lemmas and proofs in this section follow those of [1] very closely. The main difference is that tableaux and sequent proofs can split into several branches, which adds a quantifier to most of the notions and requires deciding whether something should hold for all formulae in one of the new goals or for one formula in each of the goals, etc. The other difference is that the presentation is adapted to better fit the style in which tableau/sequent calculi are usually presented.

We start with a very general notion of redundancy criterion:

Definition 3. *A* redundancy criterion *is a pair* $(\mathcal{R}_\mathcal{F}, \mathcal{R}_\mathcal{I})$ *of mappings from sets of formulae to sets of formulae, resp. sets of inferences, such that for all sets of formulae* Γ *and* Γ'*:*

(R1) *if* $\Gamma \subseteq \Gamma'$ *then* $\mathcal{R}_\mathcal{F}(\Gamma) \subseteq \mathcal{R}_\mathcal{F}(\Gamma')$*, and* $\mathcal{R}_\mathcal{I}(\Gamma) \subseteq \mathcal{R}_\mathcal{I}(\Gamma')$*.*
(R2) *if* $\Gamma' \subseteq \mathcal{R}_\mathcal{F}(\Gamma)$ *then* $\mathcal{R}_\mathcal{F}(\Gamma) \subseteq \mathcal{R}_\mathcal{F}(\Gamma \setminus \Gamma')$*, and* $\mathcal{R}_\mathcal{I}(\Gamma) \subseteq \mathcal{R}_\mathcal{I}(\Gamma \setminus \Gamma')$*.*
(R3) *if* Γ *is unsatisfiable, then so is* $\Gamma \setminus \mathcal{R}_\mathcal{F}(\Gamma)$*.*

The criterion is called effective *if, in addition,*

(R4) *an inference is in* $\mathcal{R}_\mathcal{I}(\Gamma)$*, whenever it has at least one premise introducing only formulae* $P = \{\phi_{k1}, \dots \phi_{km_k}\}$ *with* $P \subseteq \Gamma \cup \mathcal{R}_\mathcal{F}(\Gamma)$*.*

The formulae, resp. inferences in $\mathcal{R}_\mathcal{F}(\Gamma)$ *resp.* $\mathcal{R}_\mathcal{I}(\Gamma)$ *are called* redundant *with respect to* Γ*.*

For an effective redundancy criterion, any inference is redundant that has at least one premise where no new formula is introduced. This means that inferences that destroy regularity are redundant.

In contrast to resolution calculi, sequent calculi are usually written in such a way that an inference can simultaneously add new formulae and remove old ones that have become redundant. We therefore introduce the following notion:

Definition 4. *A calculus* conforms *to a redundancy criterion, if its inferences remove formulae from a branch only if they are redundant with respect to the formulae in the resulting semi-sequent.*

The following two Lemmas are taken almost verbatim from [1], where their proofs can be found.

Lemma 1. *Let* $(\Gamma_i)_{i\in\mathbb{N}}$ *be a branch of some limit derivation in a conforming calculus. Then* $\mathcal{R}_{\mathcal{F}}(\bigcup_i \Gamma_i) \subseteq \mathcal{R}_{\mathcal{F}}(\Gamma^\infty)$*, and* $\mathcal{R}_{\mathcal{I}}(\bigcup_i \Gamma_i) \subseteq \mathcal{R}_{\mathcal{I}}(\Gamma^\infty)$*.*

The next lemma is slightly different from the resolution setting, in that the implication holds in only one direction, due to the splitting into several branches.

Lemma 2. *Let* $(\Gamma_i)_{i\in\mathbb{N}}$ *be a branch of some limit derivation in a conforming calculus. If* Γ^∞ *is satisfiable, then also* Γ_0 *is satisfiable.*

We now define saturation up to redundancy which is what a derivation should approach on each branch.

Definition 5. *A set of formulae* Γ *is* saturated up to redundancy *with respect to a given calculus and redundancy criterion, if all inferences from formulae in* $\Gamma \setminus \mathcal{R}_{\mathcal{F}}(\Gamma)$ *are in* $\mathcal{R}_{\mathcal{I}}(\Gamma)$*.*

A tableau $\mathcal{T}$ *is* saturated up to redundancy *with respect to a given calculus and redundancy criterion if all its limit branches* Γ^∞ *are saturated.*

While saturation is desired for limit tableaux, the following notion gives a better idea of how a theorem prover might achieve it.

Definition 6. *A derivation* $(\mathcal{T}_i)_{i\in\mathbb{N}}$ *in a calculus that conforms to an effective redundancy criterion is called* fair *if for every limit branch* $(\Gamma_i)_{i\in\mathbb{N}}$ *of* $\mathcal{T}^\infty$*, and any non-redundant inference possible on non-redundant formulae in* Γ^∞*, all formulae of at least one of the premises of the inference are either in* $\bigcup_i \Gamma_i$ *or redundant in* $\bigcup_i \Gamma_i$*.*

Theorem 1. *If a derivation in a calculus that conforms to an effective redundancy criterion is fair, then the limit tableau it produces is saturated.*

Proof. Let γ be an inference from non-redundant formulae of some limit-branch Γ^∞ of a fair derivation. Due to fairness, for at least one premise produced by γ, $P \subseteq \bigcup_i \Gamma_i \cup \mathcal{R}_{\mathcal{F}}(\bigcup_i \Gamma_i)$, where P are the formulae γ introduces on that premise. According to (R4), γ is redundant in $\bigcup_i \Gamma_i$, and due to Lemma 1 also in Γ^∞. □

We will now make our discussion more concrete by defining a *standard redundancy criterion* which is sufficient to prove completeness of most calculi. We will prove in Theorems 2 and 3 that this standard redundancy criterion is indeed an effective redundancy criterion according to Def. 3 under certain conditions. To define the criterion, we require a fixed Noetherian order $\succ$ on formulae. We place the restriction on this ordering that $\bot$ must be smaller than all other formulae.

Definition 7. *The standard redundancy criterion is defined as follows: A formula* ϕ *is redundant with respect to a set of formulae* Γ*, iff there are formulae* $\phi_1, \ldots, \phi_n \in \Gamma$*, such that* $\phi_1, \ldots, \phi_n \models \phi$ *and* $\phi \succ \phi_i$ *for* $i = 1, \ldots, n$*.*

An inference with main formula ϕ *and side formulae* $\phi_1, \ldots \phi_n$ *is redundant w.r.t. a set of formulae* Γ*, iff it has one premise such that for all formulae* ξ *introduced in that premise, there are formulae* $\psi_1, \ldots, \psi_m \in \Gamma$*, such that* $\psi_1, \ldots, \psi_m, \phi_1, \ldots, \phi_n \models \xi$ *and* $\phi \succ \psi_i$ *for* $i = 1, \ldots, m$*.*

Theorem 2. *The standard redundancy criterion of Def. 7 is indeed a redundancy criterion according to Def. 3.*

Proof. Property (R1) follows directly from Def. 7. For property (R2), if $\phi \in \mathcal{R}_{\mathcal{F}}(\Gamma)$, consider all finite sets $\Gamma_0 \subseteq \Gamma$ of formulae smaller than ϕ which imply ϕ. Every finite set can be considered a multiset, so they can be ordered according to the multiset extension of $\succ$. Take a minimal such set. No element of Γ_0 can be redundant in Γ, since otherwise it could be replaced by some even smaller elements of Γ, contradicting the minimality of Γ_0. Therefore $\Gamma_0 \subseteq \Gamma \setminus \mathcal{R}_{\mathcal{F}}(\Gamma)$, which means that $\phi \in \mathcal{R}_{\mathcal{F}}(\Gamma \setminus \mathcal{R}_{\mathcal{F}}(\Gamma))$, and since this holds for arbitrary redundant ϕ,

$$\mathcal{R}_{\mathcal{F}}(\Gamma) \subseteq \mathcal{R}_{\mathcal{F}}(\Gamma \setminus \mathcal{R}_{\mathcal{F}}(\Gamma)) \tag{$*$}$$

To show the $\mathcal{R}_{\mathcal{F}}$ part of (R2), let $\Gamma' \subseteq \mathcal{R}_{\mathcal{F}}(\Gamma)$. This implies that $\Gamma \setminus \mathcal{R}_{\mathcal{F}}(\Gamma) \subseteq \Gamma \setminus \Gamma'$. From (R1) we get $\mathcal{R}_{\mathcal{F}}(\Gamma \setminus \mathcal{R}_{\mathcal{F}}(\Gamma)) \subseteq \mathcal{R}_{\mathcal{F}}(\Gamma \setminus \Gamma')$, and together with ($*$), $\mathcal{R}_{\mathcal{F}}(\Gamma) \subseteq \mathcal{R}_{\mathcal{F}}(\Gamma \setminus \Gamma')$. For the $\mathcal{R}_{\mathcal{I}}$ part of (R2), we consider a premise where every new formula ξ is implied by the side formulae and some formulae smaller than ϕ. The same argument as for $\mathcal{R}_{\mathcal{F}}$ can be applied to each of these ξ.

For (R3), we just showed that every redundant formula $\phi \in \mathcal{R}_{\mathcal{F}}(\Gamma)$ is implied by some non-redundant ones. Therefore $\Gamma \setminus \mathcal{R}_{\mathcal{F}}(\Gamma) \models \mathcal{R}_{\mathcal{F}}(\Gamma)$, from which (R3) follows. □

No inference in a calculus conforming to this redundancy criterion can remove $\bot$ from a semi-sequent, since $\bot$, as the smallest formula, is not redundant with respect to any set of formulae. In other words, the literal $\bot$ is always persistent. Under the following restriction, the standard redundancy criterion is *effective:*

Definition 8. *A calculus is called* reductive *if all new formulae introduced by an inference are smaller than the main formula of the inference.*

Theorem 3. *The standard redundancy criterion is an effective redundancy criterion for any reductive calculus.*

Proof. Let an inference with main formula ϕ introduce a formula $\xi \in \Gamma \cup \mathcal{R}_{\mathcal{F}}(\Gamma)$ on some premise. In a reductive calculus, $\phi \succ \xi$. If $\xi \in \Gamma$, then ξ is itself a formula smaller than ϕ which implies ξ. If $\xi \in \mathcal{R}_{\mathcal{F}}(\Gamma)$, then ξ is implied by formulae in Γ which are smaller than ξ and therefore also smaller than ϕ. If this is the case for all formulae introduced in one premise of the inference, that inference is redundant according to Def. 7. □

For the following concept, we assume a fixed *model functor* I, which maps any saturated[2] set of formulae Γ that does not contain $\bot$ to a model $I(\Gamma)$, as well as a fixed Noetherian order $\succ$ on formulae.

Definition 9. *Let Γ be saturated up to redundancy with respect to some redundancy criterion. A* counterexample for $I(\Gamma)$ in Γ *is a formula $\phi \in \Gamma$ with $I(\Gamma) \not\models \phi$. Since $\succ$ is Noetherian, if there is a counterexample for $I(\Gamma)$ in Γ, then there is also a minimal one.*

[2] This is a slight enhancement to the presentation of Bachmair and Ganzinger, who require the model functor to be defined on any (multi-)set. Knowing that the set is saturated can make it easier to define a suitable model in some cases.

A calculus has the counterexample reduction property*, if for any saturated Γ not containing $\bot$ and minimal counterexample ϕ, the calculus permits an inference*

$$\frac{\phi_{11},\ldots,\phi_{1m_1},\Gamma_0 \vdash \quad \cdots \quad \phi_{n1},\ldots,\phi_{nm_n},\Gamma_0 \vdash}{\phi,\phi_{01},\ldots,\phi_{0m_0},\Gamma_0 \vdash}$$

with main formula ϕ where $\Gamma = \{\phi,\phi_{01},\ldots,\phi_{0m_0}\} \cup \Gamma_0$ such that $I(\Gamma)$ satisfies all side formulae, i.e. $I(\Gamma) \models \phi_{01},\ldots,\phi_{0m_0}$, and each of the premises contains an even smaller counterexample ϕ_{ik_i}, i.e. $I(\Gamma) \not\models \phi_{ik_i}$ and $\phi \succ \phi_{ik_i}$.

The following lemma is similar in purpose to the usual 'model lemma' in a Hintikka-style completeness proof.

Lemma 3. *Given a calculus that*

- *conforms to the standard redundancy criterion, and*
- *is reductive, and*
- *has the counterexample reduction property,*

any set of formulae Γ that is saturated up to redundancy w.r.t. that calculus and the standard redundancy criterion, and that does not contain $\bot$, is satisfiable, specifically, $I(\Gamma) \models \Gamma$.

Proof. If the model $I(\Gamma)$ is not a model for Γ, then Γ contains a minimal counterexample ϕ. This ϕ cannot be redundant w.r.t. Γ since it would otherwise have to be a logical consequence of formulae smaller than ϕ in Γ, and all such formulae are satisfied by $I(\Gamma)$. Since the calculus has the counterexample reduction property, there is an inference with main formula ϕ, and $I(\Gamma)$ satisfying all side formulae $\phi_1,\ldots,\phi_n$, which produces a smaller counterexample ϕ' on each new premise. Since Γ is saturated, this inference must be redundant. This means that the inference has one premise, such that for the smaller counterexample ϕ' in that premise (like for all other introduced formulae), there are formulae $\psi_1,\ldots,\psi_m \in \Gamma$, all smaller than ϕ, with $\psi_1,\ldots,\psi_m,\phi_1,\ldots,\phi_n \models \phi'$. Since the ψ_i are smaller than ϕ, they too are valid in $I(\Gamma)$, and so $I(\Gamma) \models \phi'$, so ϕ' cannot be a counterexample after all. We conclude that $I(\Gamma)$ *is* a model for Γ. □

Theorem 4. *If a calculus*

- *conforms to the standard redundancy criterion, and*
- *is reductive, and*
- *has the counterexample reduction property, then*

any fair derivation for an unsatisfiable formula ϕ contains a closed tableau.

Proof. Assume that there is a fair derivation $\mathcal{T}_0,\mathcal{T}_1,\mathcal{T}_2,\ldots$ with a limit $\mathcal{T}^\infty$, where none of the $\mathcal{T}_i$ is closed. $\mathcal{T}^\infty$ has at least one branch $(\Gamma_i)_{i\in\mathbf{N}}$ that does not contain $\bot$. For assume that all limit branches contain $\bot$. These persistent formulae were introduced by some inferences in the sequence $(\mathcal{T}_i)$. Make a new tableau $\mathcal{T}'$ by cutting off every branch below the introduction of a $\bot$ literal. Then $\mathcal{T}'$ has only branches of finite length and is finitely branching. Thus, by

König's Lemma, $\mathcal{T}'$ must be a finite closed tableau for ϕ. One of the tableaux $\mathcal{T}_i$ must contain $\mathcal{T}'$ as initial sub-tableau, and thus $\mathcal{T}_i$ is closed, contradicting the assumption that there is no closed tableau in the derivation.

Now consider such an open limit branch $(\Gamma_i)_{i\in\mathbb{N}}$ with persistent formulae $\Gamma^\infty \not\ni \bot$. Due to fairness, Γ^∞ is saturated. Lemma 3 tells us that Γ^∞ is satisfiable, and due to Lemma 2, also the initial sequent Γ_0 and with it ϕ is satisfiable, contradicting our assumptions. $\square$

4 Case Study: Smullyan style NNF Tableaux

We will start by studying a familiar calculus, namely a semi-sequent calculus for first-order formulae in negation normal form (NNF). Completeness of this calculus can easily be shown with a Hintikka-style proof, but it is also a good introductory example for our new technique.

$$\alpha\ \frac{\phi, \psi, \Gamma \vdash}{\phi \wedge \psi, \Gamma \vdash} \qquad \beta\ \frac{\phi, \Gamma \vdash \qquad \psi, \Gamma \vdash}{\phi \vee \psi, \Gamma \vdash}$$

$$\gamma\ \frac{[x/t]\phi, \forall x.\phi, \Gamma \vdash}{\forall x.\phi, \Gamma \vdash} \quad \text{for any ground term } t \qquad \delta\ \frac{[x/c]\phi, \Gamma \vdash}{\exists x.\phi, \Gamma \vdash} \quad \text{for some new constant } c$$

$$\textsc{close}\ \frac{\bot \vdash}{L, \neg L, \Gamma \vdash}$$

In the CLOSE rule, we consider $\neg L$ to be the main formula, and L a side formula, since $\neg L$ is always larger than $\bot$. For the model functor, we take the set of all ground terms as domain, and we define that $I(\Gamma) \models L$ exactly for positive literals $L \in \Gamma$. We let $\succ$ order formulae by the number of boolean connectives and quantifiers appearing.

The calculus conforms to the standard redundancy criterion, since for each of the rules, the formulae deleted from the semi-sequent are clearly implied by the remaining ones. In particular for the CLOSE rule, the false formula $\bot$ implies any other formula. For the γ rule, the new formula $[x/t]\phi$ does *not* imply the original $\forall x.\phi$, but this is not required, since the original formula is kept.

The calculus is also reductive, since all rules introduce only formulae smaller than the respective main formula. Moreover, the calculus has the counterexample reduction property. For assume that $I(\Gamma) \not\models \phi$ for some $\phi \in \Gamma$. If $\phi = \phi_1 \wedge \phi_2$ is a conjunction, this means that $I(\Gamma)$ does not satisfy one of the conjuncts, w.l.o.g. ϕ_1. An α inference on ϕ is possible which produces ϕ_1, which is smaller than ϕ.

If $\phi = \phi_1 \vee \phi_2$ is a disjunction, then $I(\Gamma)$ fails to satisfy both disjuncts, and therefore each of the premises produced by the β rule contains a smaller counterexample.

In the case of a universally quantified formula, $\phi = \forall x.\phi_1$, there has to be some term t such that $I(\Gamma) \not\models [x/t]\phi_1$. The γ rule can be used to introduce

$[x/t]\phi_1$, and clearly $\forall x.\phi_1 \succ [x/t]\phi_1$, so we have reduced the counterexample. For an existentially quantified formula, $I(\Gamma) \not\models \exists x.\phi_1$, in particular $I(\Gamma) \not\models [x/c]\phi_1$, so $[x/c]\phi_1$ is a smaller counterexample.

ϕ cannot be a positive literal, since $I(\Gamma)$ is defined to satisfy all positive literals. Finally, if $\phi = \neg L$ is a negative literal, then $I(\Gamma) \models L$, and therefore $L \in \Gamma$. This allows an application of the CLOSE rule, which produces the smaller counterexample $\bot$. Note that $I(\Gamma)$ does indeed satisfy the side formula L.

Thus, Theorem 4 allows us to conclude that this calculus is complete for first order formulae in negation normal form.

5 Case Study: NNF Hyper-Tableaux

We now consider a negation normal form (NNF) version of the hyper-tableaux calculus. See [6] for an explanation of how this calculus relates to the clausal hyper-tableau calculus.

We will use the concept of *disjunctive paths* (d-paths) through formulae. The set of d-paths of a formula ϕ, denoted $dp(\phi)$, is defined by induction over the structure of ϕ as follows.

- If ϕ is a literal or a quantified formula $\forall x.\phi_1$ or $\exists x.\phi_1$, then $dp(\phi) := \{\langle\phi\rangle\}$.
- If $\phi = \phi_1 \wedge \phi_2$ is a conjunction, then $dp(\phi) := dp(\phi_1) \cup dp(\phi_2)$.
- If $\phi = \phi_1 \vee \phi_2$ is a disjunction, then $dp(\phi) := \{uv \mid u \in dp(\phi_1), v \in dp(\phi_2)\}$, where uv is the concatenation of two paths u and v.

For instance, for the formula $\phi = (p \wedge \neg p) \vee (q \wedge \neg q)$, this definition gives:

$$\begin{aligned} dp(p \wedge \neg p) &= \{\langle p\rangle, \langle\neg p\rangle\} \\ dp(q \wedge \neg q) &= \{\langle q\rangle, \langle\neg q\rangle\} \\ dp(\phi) &= \{\langle p, q\rangle, \langle p, \neg q\rangle, \langle\neg p, q\rangle, \langle\neg p, \neg q\rangle\} \end{aligned}$$

Note that we do not consider paths below quantifiers, in order to keep our discussion as simple as possible. A *positive d-path* is a d-path that contains no negated literal. In the example, $\langle p, q\rangle$ is the only positive d-path. Any d-path that is not positive must contain at least one negated literal, and in particular a left-most one. Let $lmn(\phi)$ be the set of left-most negated literals of the d-paths of ϕ. In the example, $lmn(\phi) = \{\neg q, \neg p\}$.

Consider the following semi-sequent calculus for NNF formulae:

$$\alpha\ \frac{\phi, \psi, \Gamma \vdash}{\phi \wedge \psi, \Gamma \vdash} \qquad \textsc{close}\ \frac{\bot \vdash}{L, \neg L, \Gamma \vdash}$$

$$\gamma\ \frac{[x/t]\phi, \forall x.\phi, \Gamma \vdash}{\forall x.\phi, \Gamma \vdash} \qquad \delta\ \frac{[x/c]\phi, \Gamma \vdash}{\exists x.\phi, \Gamma \vdash}$$

for any ground term t for some new constant c

$$\beta\ \frac{\phi, \Gamma \vdash \qquad \psi, \Gamma \vdash}{\phi \vee \psi, \Gamma \vdash}$$

where $\phi \vee \psi$ has at least one positive d-path.

$$\textsc{simp}\ \frac{L, \phi[L], \Gamma \vdash}{L, \phi, \Gamma \vdash}$$
where $\neg L \in lmn(\phi)$.

In the CLOSE rule and the SIMP rule, L is a side formula. By $\phi[L]$ we denote the result of replacing the negative literal $\neg L$ by the falsum $\bot$, and then simplifying the formula by repeated application of the transformations $\phi \wedge \bot \Rightarrow \bot$ and $\phi \vee \bot \Rightarrow \phi$. For instance, $((\neg p \wedge q) \vee r)[p] \Rightarrow (\bot \wedge q) \vee r \Rightarrow \bot \vee r \Rightarrow r$.

We again define the model functor, such that the domain of $I(\Gamma)$ consists of all ground terms, and $I(\Gamma) \models L$ for a positive literal L, exactly if $L \in \Gamma$. We will also use the same ordering as in the previous section.

The α, β, γ, δ, and CLOSE rules conform to the standard redundancy criterion as before. In the SIMP rule, the formula ϕ is dropped. Since $\neg L \in lmn(\phi)$, the simplification to $\phi[L]$ is indeed going to reduce the size of the formula, and an induction over the transformation steps leading to $\phi[L]$ easily convinces us that indeed $L, \phi[L] \models \phi$, so ϕ is redundant in the new sequent.

Also, the calculus is clearly still reductive, since if $\neg L \in lmn(\phi)$, then $\phi \succ \phi[L]$. As for the counterexample reduction property, the arguments are the same as in the previous section for literals, conjunctions, and quantified formulae. For disjunctions, there are two cases. If $\phi = \phi_1 \vee \phi_2$ has at least one positive d-path, a β inference will reduce the counterexample as before. Otherwise, every d-path of ϕ contains some negative literal. Then, there are again two cases: In the first case, Γ contains some L with $\neg L \in lmn(\phi)$. In particular, L is then smaller than ϕ, and therefore $I(\Gamma) \models L$. Similarly as before, we can convince ourselves that then $I(\Gamma) \not\models \phi[L]$, so the SIMP rule produces a smaller counterexample. In the second case, there is no $L \in \Gamma$ with $\neg L \in lmn(\phi)$. Due to our definition of I this means that every left-most negative literal is satisfied by $I(\Gamma)$, and therefore every d-path of ϕ contains at least one satisfied literal. Now a simple induction on the structure of ϕ, taking into account the definition of d-paths, tells us that also $I(\Gamma) \models \phi$ contradicting the assumption that ϕ is a counterexample.

Theorem 4 now tells us that also this calculus is complete.

6 Free Variable Tableaux

In this section, we shall lift the presented technique to a certain type of free variable tableaux, namely *constrained formula* free variable tableaux.They differ from the tableaux we have considered until now in two ways:

First, the formulae in the semi-sequents may contain free variables, although the formulae in the initial Γ_0 shouldn't. Free variables are used as placeholders for instantiations that a theorem prover would otherwise have to guess.

Second, semi-sequents actually contain *constrained formulae* $\phi \ll C$ consisting of a formula ϕ and a *constraint* C. For our purposes, the constraint is a formula of a subset of first order logic that will be interpreted over the domain of ground terms, using fixed interpretations for any predicate symbols. For instance in some calculi, the constraint language might be restricted to conjunctions of equations, written $s \equiv t$ denoting the syntactic equality of terms, in other cases, disjunction and negation or even quantifiers may be allowed in the constraint language, or ordering constraints $s > t$ may be available to compare terms with respect to some term ordering. For us, it is only important that there is a function Sat which for any constraint produces the set of ground substitutions that satisfy the constraint. For instance, one will usually have[3]

$$\begin{aligned} \mathrm{Sat}(s \equiv t) &= \{\sigma \in \mathcal{G} \mid \sigma s = \sigma t\} \\ \mathrm{Sat}(C \mathbin{\&} D) &= \mathrm{Sat}(C) \cap \mathrm{Sat}(D) \\ \mathrm{Sat}(!\, C) &= \mathcal{G} \setminus \mathrm{Sat}(C) \end{aligned} \tag{†}$$

etc., so & denotes conjunction of constraints, ! negation of constraints, where $\mathcal{G}$ is the set of all ground substitutions.

A constrained formula $\phi \ll C$ in a semi-sequent means that the formula has resulted from some sequence of inferences that are only sound (or more generally desired) in cases where the free variables get instantiated as described by C. Ultimately, the constraints will get propagated until they reach $\bot \ll C$, of which a suitable combination has to be found to close all branches of the proof.

Definition 10. *A tableau of a constrained formula tableau calculus is* closed under σ*, where* σ *is a ground substitution for the occurring free variables, iff every leaf sequent of the tableau contains a constrained formula* $\phi \ll C$ *with* $\sigma \in \mathrm{Sat}(C)$*. A tableau is* closable *if there exists a* σ *under which it is closed.*

The following definition describes how to apply a substitution to a semi-sequent or to a whole tableau, while discarding any formulae which carry a constraint that is not satisfied by that substitution.

Definition 11. *Let* Γ *be a set of constrained formulae. We define*

$$\sigma\Gamma := \{\sigma\phi \mid \phi \ll C \in \Gamma \text{ with } \sigma \in \mathrm{Sat}(C)\} \quad .$$

Let $\mathcal{T}$ *be a tableau. We construct* $\sigma\mathcal{T}$ *by replacing the semi-sequent* Γ *in each node of* $\mathcal{T}$ *by* $\sigma\Gamma$*.*

The next definition uses these notions of substitution to establish a tight correspondence between constrained variable calculi and the non-free-variable, non-constrained calculi described in the previous sections. The correspondence is actually the same as that between Smullyan-style first order tableaux and Fitting-style [4] free variable tableaux.

Definition 12. *Let*

$$\frac{\Gamma_1 \vdash \quad \cdots \quad \Gamma_n \vdash}{\Gamma_0}$$

be an inference of a constrained formula tableau calculus. The corresponding ground inference under σ *for some ground substitution* σ *is*

$$\frac{\sigma\Gamma_1 \vdash \quad \cdots \quad \sigma\Gamma_n \vdash}{\sigma\Gamma_0} \quad .$$

[3] We use the prenex notation "σt", etc., for the application of substitutions.

The corresponding ground calculus *is the calculus consisting of all corresponding ground inferences under any σ of any inferences in the constrained formula calculus.*

We use the word 'ground' for notions without free variables. There might well be quantifiers in the formulae involved. Any given inference of the constrained formula calculus can in general have infinitely many different corresponding ground inferences for different σ, but each of them is an ordinary, finite ground inference.

Using the correspondence to a ground calculus, we can define the same properties as before for constrained formula tableaux:

Definition 13. *A constrained formula calculus* conforms *to a given redundancy criterion, has the* counterexample reduction property*, or is* reductive *iff the corresponding ground calculus has that property.*

Note that a constrained formula calculus can always discard a formula with an unsatisfiable constraint, since it disappears under any substitution. Therefore, a ground inference corresponding to the deletion of a formula with unsatisfiable constraint does not change the semi-sequent, so it trivially conforms to any redundancy criterion.

Finally, a notion of fairness is needed. Again this definition is heavily based on the 'ground' notion. We will discuss its implications after Theorem 5.

Definition 14. *A constrained formula tableau derivation $(\mathcal{T}_i)_{i\in\mathbf{N}}$ in a calculus that conforms to an effective redundancy criterion is called* fair *if there is a ground substitution σ for the free variables, such that $(\sigma\mathcal{T}_i)_{i\in\mathbf{N}}$ is a fair derivation of the corresponding ground calculus. We call such a σ a* fair instantiation *for the constrained formula tableau derivation.*

It is now easy to show completeness for well-behaved calculi:

Theorem 5. *If a constrained formula calculus*

- *conforms to the standard redundancy criterion and*
- *is reductive*
- *has the counterexample reduction property, then*

a fair derivation for an unsatisfiable formula ϕ contains a closable tableau.

Proof. Let σ be a fair instantiation for $(\mathcal{T}_i)_{i\in\mathbf{N}}$. Then $(\sigma\mathcal{T}_i)_{i\in\mathbf{N}}$ is a fair derivation of the corresponding ground calculus and $\sigma\mathcal{T}_0 = \mathcal{T}_0$, since the initial formula ϕ does not contain free variables. Theorem 4 ensures that some $\sigma\mathcal{T}_i$ is closed. Therefore, $\mathcal{T}_i$ is closed under σ. □

The big question is of course whether a constrained formula calculus actually admits fair derivations, and how these can be constructed algorithmically. There are two issues to be discussed here.

The first is that a series of inferences on a branch might 'change' the constraint of a formula ϕ, successively deriving $\phi \ll C_0$, $\phi \ll C_1, \ldots$. None of these formulae is persistent in the usual sense of the word. Still, there can be a substitution $\sigma \in \mathrm{Sat}(C_0) \cap \mathrm{Sat}(C_1) \cap \cdots$. The *instantiation* $\sigma\phi$ is therefore persistent, and a fair derivation must eventually perform inferences that correspond to ground inferences on $\sigma\phi$.

Consider for instance a calculus with the following hypothetical rules:

$$\textsc{step}\ \frac{p(t) \ll A, p(f(t)) \ll A, \Gamma \vdash}{p(t) \ll A, \Gamma \vdash}\ \text{for any term } t \qquad \textsc{close}\ \frac{\bot \ll A \vdash}{r(t) \ll A, \Gamma \vdash}\ \text{for any term } t$$

$$\textsc{reduce}\ \frac{q(s) \ll A \,\&\, B \,\&\, s \equiv t, r(s) \ll A \,\&\, !(B \,\&\, s \equiv t), p(t) \ll B, \Gamma \vdash}{r(s) \ll A, p(t) \ll B, \Gamma \vdash}\ \text{for any terms } s,\ t$$

where & is conjunction and ! is negation of constraints, and $\equiv$ denotes syntactic equality, as described by the equations (†). From a sequent

$$r(X),\ p(a) \vdash$$

we can derive, using REDUCE

$$q(X) \ll X \equiv a,\ r(X) \ll !\, X \equiv a,\ p(a) \vdash$$

and then, with STEP,

$$q(X) \ll X \equiv a,\ r(X) \ll !\, X \equiv a,\ p(a),\ p(f(a)) \vdash \quad .$$

Now we apply REDUCE again:

$$q(X) \ll X \equiv f(a),\ q(X) \ll X \equiv a,\ r(X) \ll !\, X \equiv a \& !\, X \equiv f(a),\ p(a),\ p(f(a) \vdash$$

and so on. The constraint on $r(X)$ gets more and more complicated, and none of the constrained formulae is persistent. But for fairness, we must eventually apply CLOSE, since this will not become redundant whatever the instantiation for X (unless there are other rules which eventually close the branch).

For some calculi, like standard free variable tableaux without constraints, such situations simply cannot occur. If they can however, a possible solution is to use a theorem proving procedure that achieves fairness not by managing a queue of formulae that remain to be processed, but a queue of rule applications: Any new constrained formula introduced to a branch should be checked for possible inferences in combination with other present formulae. All possible inferences should eventually be considered in a fair manner, *even if the original constrained formula gets deleted or changed*. When an inference's turn has come, it should be checked whether there are *now* formulae in the semi-sequent on which it can be applied.

The second issue is that of the fair instantiation of free variables. In many calculi, free variables are only introduced by a γ rule like[4]

$$\gamma \; \frac{[x/X]\phi, \forall x.\phi, \Gamma \vdash}{\forall x.\phi, \Gamma \vdash}$$

with a new free variable X.

The corresponding ground inferences are

$$\frac{[x/t]\phi, \forall x.\phi, \Gamma \vdash}{\forall x.\phi, \Gamma \vdash}$$

for any term t. In general, if a formula $\forall x.\phi$ is persistent on some branch of a fair ground derivation, this rule needs to be applied for *all* ground terms t, with the possible exception of terms for which $[x/t]\phi$ happens to be redundant. Therefore, in the calculus with free variables, a fair instantiation can in general only exist if infinitely many copies of $[x/X_i]\phi$ with different free variables are introduced on the branch. Therefore, the theorem proving procedure has to apply the γ rule again and again. The fair instantiation can then be defined by taking for instance an enumeration $(t_i)_{i\in\mathbf{N}}$ of all ground terms and requiring that $\sigma X_i = t_i$.

This is not necessarily the case for every rule that introduces a free variable. For instance, in Sect. 6 of [5], a constrained formula tableau version of the basic ordered paramodulation rule [2] is given, in which the new free variable is constrained to only one possible instantiation. Therefore, this rule needs to be applied only once.

Although these observations about fairness should cover the most common cases, in the framework given so far, the question ultimately has to be considered for each calculus. It will be an interesting topic for future research to find sensible restrictions of the given framework that permit general statements about fairness.

Another general remark is in order concerning our 'lifting', i.e. the relation between our free variable calculi to ground calculi. In particular for equality handling by rewriting, it is important to restrict the application of equalities to non-variable positions. This means that an inference that acts only on the instantiation of some free variables should not be needed. The framework presented so far does not help in excluding such inferences. A corresponding refinement is a further topic for future research.

7 Case Study: Free Variable NNF Hyper-Tableaux

We will now study a constrained formula version of the NNF hyper-tableaux calculus of Sect. 5. Completeness of such a calculus has previously been shown using proof transformation arguments [6], but the proof using saturation up to

---

[4] When we don't write constraints, we mean the trivial constraint that is satisfied by all instantiations of the free variables.

redundancy will be a lot simpler. We will start from pre-skolemized formulae that contain no existential quantifiers, to avoid discussing the soundness issues that arise in connection with free variables in δ rules.

The rules of our calculus are as follows:

$$\alpha\ \frac{\phi \ll A,\ \psi \ll A, \Gamma \vdash}{\phi \wedge \psi \ll A, \Gamma \vdash} \qquad \beta\ \frac{\phi \ll A, \Gamma \vdash \quad \psi \ll A, \Gamma \vdash}{\phi \vee \psi \ll A, \Gamma \vdash}$$

where $\phi \vee \psi$ has at least one positive d-path.

$$\gamma\ \frac{[x/X]\phi \ll A, \forall x.\phi \ll A, \Gamma \vdash}{\forall x.\phi \ll A, \Gamma \vdash} \qquad \textsc{close}\ \frac{\bot \ll L \equiv M \,\&\, A \,\&\, B \vdash}{L \ll A,\ \neg M \ll B, \Gamma \vdash}$$

for a new free variable X

$$\textsc{simp}\ \frac{\mu\phi[\mu L] \ll L \equiv M \,\&\, A \,\&\, B, \phi \ll A \,\&\, !(L \equiv M \,\&\, B),\ L \ll B, \Gamma \vdash}{\phi \ll A,\ L \ll B, \Gamma \vdash}$$

where $\neg M \in lmn(\phi)$ and μ is a most general unifier of L and M.

It is not hard to see that the ground instances corresponding to the α, β, γ, and CLOSE rules are exactly the inferences of the respective rules in Sect. 4. For the SIMP rule, the corresponding ground inference under some instantiation $\sigma \in \mathrm{Sat}(L \equiv M \,\&\, A \,\&\, B)$ is

$$\frac{\sigma\phi[\sigma L],\ \sigma L, \Gamma \vdash}{\sigma\phi,\ \sigma L, \Gamma \vdash}$$

which is just the SIMP rule of Sect. 5. For all $\sigma \notin \mathrm{Sat}(L \equiv M \,\&\, A \,\&\, B)$, the constraints ensure that the corresponding ground inference under σ does not change the sequent. It follows that apart from the missing δ rule, the corresponding ground calculus is exactly the one from Sect. 5.

We conclude that any proof procedure that produces fair derivations in this calculus is complete. Let us analyze what is needed for fairness: free variables can only be introduced by the γ rule, so as discussed before, it needs to be applied infinitely often on each branch for any persistent occurrence of a constrained formula $\forall x.\phi \ll C$, producing formulae $[x/X_i]\phi$ with distinct variables. Since there are no rules that could delete such an occurrence, *all* occurrences of universally quantified formulae are persistent. The corresponding fair instantiation σ needs to make sure that if $\sigma \in \mathrm{Sat}(C)$, then there is an X_i with $\sigma X_i = t$ for every ground term t. This is of course a well-known ingredient in many tableau completeness proofs.

Do we have the fairness problem of persistent ground instances $\sigma\phi$ of non-persistent formulae $\phi \ll C$ described in the previous section? Yes, we do! The SIMP rule can lead to similar chains of inferences as the REDUCE rule. Consider a formula $\phi = (\neg p(X) \wedge \neg p(b)) \vee q(X)$ in the place of the $r(X)$ in the REDUCE example. From a series of literals $p(a), p(f(a)), \ldots$, the SIMP rule allows to derive $q(a), q(f(a)), \ldots$, constantly changing the constraint on ϕ, although a SIMP on the other left-most negative literal $\neg p(b)$ might be possible all the time and necessary for completeness.

In this calculus, there is an easier way of coping with this problem than the one we hinted at in Sect. 6: Theorem 3 of [6] establishes the interesting fact that under certain sensible restrictions, our calculus always permits only a finite number of inferences without intervening γ inferences. This means that we can obtain fairness simply by requiring derivations to be built in such a way that γ inferences may only be applied when there are no more possible SIMP inferences.

This illustrates that the fairness question can be quite subtle, depending on the particular calculus at hand.

8 Conclusion

We have introduced a notion of saturation up to redundancy for tableau and sequent calculi, closely following the work of Bachmair and Ganzinger [1] for resolution calculi. We have shown a generic completeness theorem that makes it easy to show completeness of calculi with destructive rules. Notions and proofs were lifted to the case of free variable tableaux with constrained formulae. Some examples were given to illustrate the method.

Future work includes finding a generic way of achieving fairness for free variable calculi. A method of lifting that does not require inferences below variable positions would be needed to apply our technique to equality reasoning. One might also consider defining when whole branches are redundant with respect to the rest of a tableau, to allow redundancy elimination on the branch level. It might also be interesting to adapt the idea of 'histories' used in [5] instead of constraints with negations to our framework.

References

1. L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 2, pages 19–99. Elsevier Science B.V., 2001.
2. L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995.
3. D. Cantone and C. G. Zarba. A tableau calculus for integrating first-order reasoning with elementary set theory reasoning. In R. Dyckhoff, editor, *Proc. TABLEAUX 2000*, volume 1847 of *LNCS*, pages 143–159. Springer, 2000.
4. M. C. Fitting. *First-Order Logic and Automated Theorem Proving*. Springer, second edition, 1996.
5. M. Giese. A model generation style completeness proof for constraint tableaux with superposition. In U. Egly and C. G. Fermüller, editors, *Proc. TABLEAUX 2002*, volume 2381 of *LNCS*, pages 130–144. Springer, 2002.
6. M. Giese. Simplification rules for constrained formula tableaux. In M. Cialdea Mayer and F. Pirri, editors, *Proc. TABLEAUX 2003*, volume 2796 of *LNCS*, pages 65–80. Springer, 2003.
7. M. Giese. A calculus for type predicates and type coercion. In B. Beckert, editor, *Proc. TABLEAUX 2005*, volume 3702 of *LNAI*, pages 123–137. Springer, 2005.
8. R. M. Smullyan. *First-Order Logic*, volume 43 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, 1968.