Term sequent logic

Michael Gabbay[1]

http://www.kcl.ac.uk/kis/schools/hums/philosophy/staff/m_gabbay.html

Murdoch J. Gabbay

http://www.gabbay.org.uk

Abstract

We consider a term sequent logic for the lambda-calculus. Term sequents are a judgement form similar to the logical judgement form of entailment between sentences, but denoting equality or reducibility between terms. Using term sequents, it is possible to treat lambda-terms almost like logical sentences, and to use proof-theoretic methods to establish their properties. We prove a cut-elimination result for untyped lambda-calculus and describe how this generalises the usual confluence result. We give a notion of uniform proof for lambda-terms, and suggest how this can be viewed as a mixed logic-programming/functional programming framework with the ability to assume arbitrary reductions. Finally, we discuss related and future work.

Keywords: Cut Elimination, Lambda Calculus, Functional programming, Proof Theory

1 Introduction

Sequent calculi are a general framework for formalising logical consequence relations and proving their properties. Cut-elimination is key to proving properties such as consistency, non-triviality (a model exists with more than one element), non-derivability, completeness for derivation-search algorithms, decidability results for fragments, and so on. However, these properties do not follow in the presence of axioms. For example cut-elimination does not directly imply the consistency of an equality axiom like $(\lambda x.x){\cdot}y \approx y$.

Term sequents generalise sequent calculus; the intuition is 'sequents for terms, as well as sentences'. Logical consequence becomes a relation not just between sentences, but also between terms. An advantage of doing this is that a term sequent calculus represents, without axioms, logics with non-trivial equalities between terms — logics that would require axioms if formulated using the 'ordinary' sequent calculus, thus precluding an obvious proof-theoretic treatment of their properties.

[1] Michael Gabbay gratefully acknowledges the support of the British Academy under the grant PDF/2006/509.

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

$$\frac{}{\Gamma \vdash \mathtt{x} :\!- \mathtt{x}}\ (Ax_{:-}) \qquad \frac{\Gamma \vdash \Theta :\!- \mathtt{s} \quad \Gamma \vdash \langle \ldots \mathtt{s} \ldots \rangle :\!- \mathtt{t}}{\Gamma \vdash \langle \ldots \Theta \ldots \rangle :\!- \mathtt{t}}\ (Cut_\lambda)$$

$$\frac{\Gamma \vdash \langle \ldots \langle \mathtt{t}_1, \mathtt{t}_2 \rangle \ldots \rangle :\!- \mathtt{t}}{\Gamma \vdash \langle \ldots \mathtt{t}_1 {\cdot} \mathtt{t}_2 \ldots \rangle :\!- \mathtt{t}}\ (\cdot L) \qquad \frac{\Gamma \vdash \Theta_1 :\!- \mathtt{t}_1 \quad \Gamma \vdash \Theta_2 :\!- \mathtt{t}_2}{\Gamma \vdash \langle \Theta_1, \Theta_2 \rangle :\!- \mathtt{t}_1 {\cdot} \mathtt{t}_2}\ (\cdot R)$$

$$\frac{\Gamma \vdash \Theta :\!- \mathtt{t}_1 \quad \Gamma \vdash \langle \ldots \mathtt{t}_2 \ldots \rangle :\!- \mathtt{t}}{\Gamma, \mathtt{t}_1 \rightsquigarrow \mathtt{t}_2 \vdash \langle \ldots \Theta \ldots \rangle :\!- \mathtt{t}}\ (\rightsquigarrow L) \qquad \frac{\Gamma \vdash \mathtt{t}_1 :\!- \mathtt{t}_2}{\Gamma \vdash \mathtt{t}_1 \rightsquigarrow \mathtt{t}_2, \Delta}\ (\rightsquigarrow R)$$

$$\frac{\Gamma \vdash \Theta :\!- \mathtt{t}_1 \quad \Gamma \vdash \langle \ldots \mathtt{t}_2[\mathtt{x}/\mathtt{t}_1] \ldots \rangle :\!- \mathtt{t}}{\Gamma \vdash \langle \ldots \langle \lambda \mathtt{x}.\mathtt{t}_2, \Theta \rangle \ldots \rangle :\!- \mathtt{t}}\ (\lambda L) \qquad \frac{\Gamma \vdash \langle \Theta, \mathtt{x} \rangle :\!- \mathtt{t}}{\Gamma \vdash \Theta :\!- \lambda \mathtt{x}.\mathtt{t}}\ (\lambda R)\ \ \text{x not free in } \Theta \text{ or } \Gamma$$

Fig. 1. Term sequent rules: λ-calculus

$$\frac{\Gamma, \mathtt{A}_i \vdash \Delta}{\Gamma, \mathtt{A}_1 \wedge \mathtt{A}_2 \vdash \Delta}\ (\wedge L)\ \ 1 \leq i \leq 2 \qquad \frac{\Gamma \vdash \mathtt{A}_1, \Delta \quad \Gamma \vdash \mathtt{A}_2, \Delta}{\Gamma \vdash \mathtt{A}_1 \wedge \mathtt{A}_2, \Delta}\ (\wedge R) \qquad \frac{\Gamma \vdash \mathtt{C}, \Delta \quad \Gamma, \mathtt{C} \vdash \Delta}{\Gamma \vdash \Delta}\ (Cut)$$

$$\frac{\Gamma, \mathtt{A}[\mathtt{x}/\mathtt{t}] \vdash \Delta}{\Gamma, \forall \mathtt{x}.\mathtt{A} \vdash \Delta}\ (\forall L) \qquad \frac{\Gamma \vdash \mathtt{A}, \Delta}{\Gamma \vdash \forall \mathtt{x}.\mathtt{A}, \Delta}\ (\forall R)\ \ \text{x not free in } \Gamma, \Delta \qquad \frac{\Gamma, \mathtt{A} \vdash \Delta}{\Gamma \vdash \neg \mathtt{A}, \Delta}\ (\neg R) \qquad \frac{\Gamma \vdash \mathtt{A}, \Delta}{\Gamma, \neg \mathtt{A} \vdash \Delta}\ (\neg L)$$

$$\frac{\Gamma \vdash \mathtt{A}, \Delta \quad \Gamma, \mathtt{B} \vdash \Delta}{\Gamma, \mathtt{A} \Rightarrow \mathtt{B} \vdash \Delta}\ (\Rightarrow L) \qquad \frac{\Gamma, \mathtt{A} \vdash \mathtt{B}, \Delta}{\Gamma \vdash \mathtt{A} \Rightarrow \mathtt{B}, \Delta}\ (\Rightarrow R)$$

Fig. 2. Sentence sequent rules

As a result, the basic technique of cut-elimination is extended to algebraic systems (those satisfying non-trivial equalities between terms).

The specific objective of this paper is to develop a term-sequent style logic and proof-theory of the (untyped) λ-calculus. We shall prove full cut-elimination and also exhibit a well-behaved notion of uniform proof. This reconciles, in a novel way, first-order logic with the λ-calculus, and the computational content of first-order logic with the computational content of the λ-calculus.

The idea of logics with term sequents is itself novel to this paper. We have applied it to the λ-calculus, but first author has developed term sequent systems for other systems, for example arithmetic and rational numbers. Further comment is in the Conclusions.

2 Syntax of the λ-calculus

Definition 2.1 – Define ***terms*** $\mathtt{t}$ and ***sentences*** $\mathtt{A}$ by:

$$\mathtt{t} ::= \mathtt{x}, \mathtt{y}, \mathtt{z}, \ldots \mid (\mathtt{t}_1 {\cdot} \mathtt{t}_2) \mid (\lambda \mathtt{x}.\mathtt{t})$$

$$\mathtt{A} ::= \mathtt{t}_1 \rightsquigarrow \mathtt{t}_2 \mid \mathtt{A}_1 \wedge \mathtt{A}_2 \mid \mathtt{A}_1 \Rightarrow \mathtt{A}_2 \mid \neg \mathtt{A} \mid \forall \mathtt{x}.\mathtt{A}$$

– Free variables of $\mathtt{t}$ and $\mathtt{A}$ are defined as usual; for example $\mathtt{x}$ is free in $\mathtt{x} \rightsquigarrow \mathtt{y}$ and

$$(D1)\ \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \mathtt{t}[\mathtt{x}/\mathtt{y}] \mathrel{:\!-} \mathtt{t}[\mathtt{x}/\mathtt{y}]}\,\text{Thrm. 3.1}}{\vdash \langle\lambda\mathtt{x}.\mathtt{t},\mathtt{y}\rangle \mathrel{:\!-} \mathtt{t}[\mathtt{x}/\mathtt{y}]}\,(\lambda L)}{\vdash \lambda\mathtt{x}.\mathtt{t} \mathrel{:\!-} \lambda\mathtt{y}.(\mathtt{t}[\mathtt{x}/\mathtt{y}])}\,(\lambda R)}{\vdash \lambda\mathtt{x}.\mathtt{t}\rightsquigarrow\lambda\mathtt{y}.(\mathtt{t}[\mathtt{x}/\mathtt{y}])}\,(\rightsquigarrow R)
\qquad
(D2)\ \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \mathtt{t}[\mathtt{x}/\mathtt{y}][\mathtt{y}/\mathtt{x}] \mathrel{:\!-} \mathtt{t}}\,\text{Thrm. 3.1}}{\vdash \langle\lambda\mathtt{y}.(\mathtt{t}[\mathtt{x}/\mathtt{y}]),\mathtt{x}\rangle \mathrel{:\!-} \mathtt{t}}\,(\lambda L)}{\vdash \lambda\mathtt{y}.(\mathtt{t}[\mathtt{x}/\mathtt{y}]) \mathrel{:\!-} \lambda\mathtt{x}.\mathtt{t}}\,(\lambda R)}{\vdash \lambda\mathtt{y}.(\mathtt{t}[\mathtt{x}/\mathtt{y}])\rightsquigarrow\lambda\mathtt{x}.\mathtt{t}}\,(\rightsquigarrow R)$$

$$(D3)\ \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \mathtt{t}[\mathtt{x}/\mathtt{t}'] \mathrel{:\!-} \mathtt{t}[\mathtt{x}/\mathtt{t}']}\,\text{Thrm. 3.1}}{\vdash \langle\lambda\mathtt{x}.\mathtt{t},\mathtt{t}'\rangle \mathrel{:\!-} \mathtt{t}[\mathtt{x}/\mathtt{t}']}\,(\lambda L)}{\vdash (\lambda\mathtt{x}.\mathtt{t})\cdot\mathtt{t}' \mathrel{:\!-} \mathtt{t}[\mathtt{x}/\mathtt{t}']}\,(\cdot L)}{\vdash (\lambda\mathtt{x}.\mathtt{t})\cdot\mathtt{t}'\rightsquigarrow\mathtt{t}[\mathtt{x}/\mathtt{t}']}\,(\rightsquigarrow R)
\qquad
(D4)\ \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \mathtt{t} \mathrel{:\!-} \mathtt{t}}\,\text{Thrm. 3.1}\quad \dfrac{}{\vdash \mathtt{x} \mathrel{:\!-} \mathtt{x}}\,(Ax_{:-})}{\vdash \langle\mathtt{t},\mathtt{x}\rangle \mathrel{:\!-} \mathtt{t}\cdot\mathtt{x}}\,(\cdot L)}{\vdash \mathtt{t} \mathrel{:\!-} \lambda\mathtt{x}.(\mathtt{t}\cdot\mathtt{x})}\,(\lambda R)}{\vdash \mathtt{t}\rightsquigarrow\lambda\mathtt{x}.(\mathtt{t}\cdot\mathtt{x})}\,(\rightsquigarrow R)$$

Fig. 3. Derivations of $\alpha\beta\eta$-reduction

in $\mathtt{x}\cdot\mathtt{x}$, and $\mathtt{x}$ is not free in $\forall\mathtt{x}.\mathtt{A}$ and $\lambda\mathtt{x}.(\mathtt{x}\cdot\mathtt{x})$.

– Also as usual, we take predicates and terms up to α-conversion of bound variables. We use $\equiv$ to express identity of syntax up to α-conversion.

Definition 2.2 – Define ***trees*** by: $\Theta ::= \mathtt{t} \mid \langle\Theta_1, \Theta_2\rangle$.

– If Θ' is a *subtree* of Θ then we write Θ as $\langle\ldots\Theta'\ldots\rangle$.

Definition 2.3 A ***term sequent*** is a tuple $\Gamma\vdash\Theta \mathrel{:\!-} \mathtt{t}$ where Θ is a tree and $\mathtt{t}$ is a term. A ***sentence sequent*** is a pair $\Gamma\vdash\Delta$. The ***derivable sequents*** of a term sequent logic are inductively defined by the rules in Figures 1 and 2.

For some example derivations, see Figure 3.

Remark 2.4 As a piece of design of a logic, issues lie in the choice of structure for a term sequent $\Gamma\vdash\Theta \mathrel{:\!-} \mathtt{t}$, and of term sequent derivation rules. The challenge is to find a term sequent structure and rules such that terms and their term-formers can be decomposed in a syntax-directed manner.

This design issue is familiar from other sequent systems: for example bunched implications [12] requires a logical context with 'bunches'; intuitionistic logic typically has a single sentence on the right whereas classical logic has many sentences; and so on.

Remark 2.5 An intuition for the rules of Figure 1 is perhaps best obtained by examining how they contribute to Theorem 3.6, which makes a correspondence between derivability in our term sequent logic, and $\alpha\beta\eta$-reductions in familiar formulations of the λ-calculus; it is a kind of soundness and completeness result.

In a term sequent $\Gamma\vdash\Theta \mathrel{:\!-} \mathtt{t}$, we may see Γ as a set of assumptions, $\mathtt{t}$ as a λ-term and Θ as a rough normal form of a λ-term that reduces to $\mathtt{t}$. Intuitively, (λL) corresponds to β-reduction; (λR) corresponds to η-expansion; $(\cdot L)$ fixes $\cdot$ as a symbol for application and $(\cdot R)$ corresponds to a form of congruence.

Remark 2.6 A special case of (λL) is $\dfrac{\Gamma\vdash\mathtt{t}_1 \mathrel{:\!-} \mathtt{t}_1 \quad \Gamma\vdash\langle\ldots\mathtt{t}_2[\mathtt{x}/\mathtt{t}_1]\ldots\rangle \mathrel{:\!-} \mathtt{t}}{\Gamma\vdash\langle\ldots\langle\lambda\mathtt{x}.\mathtt{t}_2,\mathtt{t}_1\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}}$. We can write this as $\dfrac{\Gamma\vdash\langle\ldots\mathtt{t}_2[\mathtt{x}/\mathtt{t}_1]\ldots\rangle \mathrel{:\!-} \mathtt{t}}{\Gamma\vdash\langle\ldots\langle\lambda\mathtt{x}.\mathtt{t}_2,\mathtt{t}_1\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}}$ in the light of Theorem 3.1.

Similarly, a special case of $(\rightsquigarrow L)$ may be written $\dfrac{\Gamma\vdash\langle\ldots\mathtt{t}_2\ldots\rangle \mathrel{:\!-} \mathtt{t}}{\Gamma,\mathtt{t}_1\rightsquigarrow\mathtt{t}_2\vdash\langle\ldots\mathtt{t}_1\ldots\rangle \mathrel{:\!-} \mathtt{t}}$.

These special cases are insufficient for term sequent logic; they fail to yield cut-elimination in the same way that this special case of $(\Rightarrow L)$ $\dfrac{\Gamma,\mathtt{B}\vdash\Delta}{\Gamma,\mathtt{A}\Rightarrow\mathtt{B},\mathtt{A}\vdash\Delta}$ fails to yield cut-elimination in propositional logic. For example $(\mathtt{A}_1 \wedge \mathtt{A}_2) \Rightarrow \mathtt{B}, \mathtt{A}_1, \mathtt{A}_2 \vdash \mathtt{C}$ is not derivable without (Cut) if only the special case of $(\Rightarrow L)$ is used. Similarly, $\vdash \langle \lambda \mathtt{y}.\mathtt{y}, \langle \mathtt{s}, \mathtt{t} \rangle \rangle :- \mathtt{s}{\cdot}\mathtt{t}$ requires (Cut_λ) if we use only the special case of (λL).

We read $\Theta = \langle \Theta_1, \Theta_2 \rangle$ as 'Θ_1 applied to Θ_2'. We read $\Gamma \vdash \Theta :- \mathtt{t}$ as: "Γ implies that the term corresponding to Θ is equal/reduces to $\mathtt{t}$." So we read $\Gamma \vdash \langle \lambda \mathtt{x}.(\mathtt{y}{\cdot}\mathtt{x}), \mathtt{z} \rangle :- \lambda \mathtt{y}.\mathtt{z}$ as "Γ implies that $\lambda \mathtt{x}.(\mathtt{y}{\cdot}\mathtt{x})$ applied to $\mathtt{z}$ reduces to $\lambda \mathtt{y}.\mathtt{z}$".

Remark 2.7 Our notation uses three different sorts of bracket, but we promise that this is harmless.

- Square brackets [ ] express capture avoiding substitution on terms, this is standard.
- Round brackets ( ) parse terms, as is standard.
- Angle brackets $\langle\ \rangle$ parse trees (Definition 2.2), which are part of our term-sequent form.

Because term sequent derivation rules break apart terms, it can happen that round brackets 'become' angle brackets (e.g. the last three lines of derivation $D3$ of Figure 3). This has essentially the same status as conjunctions 'becoming' commas in traditional sequent systems.

Remark 2.8 Manipulation of term sequents and sentence sequents are kept separate. Only $(\rightsquigarrow R)$ moves between them. The two cut rules (Cut_λ) and (Cut) apply either to terms in term sequents, or to sentences in sentence sequents. For example we cannot use our cut rules to make an inference like

$$\frac{\overline{\mathtt{A} \wedge \mathtt{t}_1 \rightsquigarrow \mathtt{t}_2 \vdash \mathtt{t}_1 \rightsquigarrow \mathtt{t}_2} \quad \overline{\mathtt{t}_1 \rightsquigarrow \mathtt{t}_2 \vdash \mathtt{t}_1 :- \mathtt{t}_2}}{\mathtt{A} \wedge \mathtt{t}_1 \rightsquigarrow \mathtt{t}_2 \vdash \mathtt{t}_1 :- \mathtt{t}_2}$$

This separation is necessary and slightly restricts deductive power, for example we cannot derive that $\mathtt{A} \wedge \mathtt{t}_1 \rightsquigarrow \mathtt{t}_2 \vdash \mathtt{t}_1 :- \mathtt{t}_2$, but it simplifies matters.

Remark 2.9 A notion similar to $\rightsquigarrow$ has been investigated under the name *aequality*, using the same symbol, and representing a directed equality in a first-order logic [4]. The intuition of $\rightsquigarrow$ here is similar — 'directed equality' is much the same thing as 'reduction'. Further intuition for $\rightsquigarrow$ is provided by in Theorem 3.6.

3 Derivability

Theorem 3.1 *– $\Gamma \vdash \mathtt{t} :- \mathtt{t}$ is derivable, without using (Cut_λ), for any $\mathtt{t}$.*
– $\Gamma, \mathtt{A} \vdash \mathtt{A}, \Delta$ is always derivable.

Proof. The first part is by induction on $\mathtt{t}$. For example, if $\mathtt{t} \equiv \lambda x.\mathtt{s}$ then by induction hypothesis $\Gamma \vdash \mathtt{s} :- \mathtt{s}$, and so:

$$\cfrac{\cfrac{\begin{matrix}\Pi \\ \Gamma \vdash \mathtt{s} :- \mathtt{s}\end{matrix}}{\Gamma \vdash \langle \lambda \mathtt{x}.\mathtt{s}, \mathtt{x} \rangle :- \mathtt{s}}\ (\lambda L)}{\Gamma \vdash \lambda.\mathtt{xs} :- \lambda \mathtt{x}.\mathtt{s}}\ (\lambda R)$$

The second part follows by $(\rightsquigarrow L)$, $(\rightsquigarrow R)$ and induction on $\mathtt{A}$. □

Theorem 3.2

$\mathtt{x}\rightsquigarrow\mathtt{y}, \mathtt{y}\rightsquigarrow\mathtt{z} \vdash \mathtt{x}\rightsquigarrow\mathtt{z}$	$\lambda\mathtt{x}.\mathtt{t} :\!- \lambda\mathtt{y}.(\mathtt{t}[\mathtt{x}/\mathtt{y}])$ *(if $\mathtt{y}$ is not free in $\mathtt{t}$)*
$\vdash \mathtt{x}\rightsquigarrow\mathtt{x}$	$\lambda\mathtt{y}.(\mathtt{t}[\mathtt{x}/\mathtt{y}]) :\!- \lambda\mathtt{x}.\mathtt{t}$ *(if $\mathtt{y}$ is not free in $\mathtt{t}$)*
$\nvdash \mathtt{x} :\!- \mathtt{y}$	$(\lambda\mathtt{x}.\mathtt{t})\cdot\mathtt{t}' :\!- \mathtt{t}[\mathtt{x}/\mathtt{t}']$
$\nvdash \mathtt{x}\rightsquigarrow\mathtt{y}$	$\mathtt{t} :\!- \lambda\mathtt{x}.(\mathtt{t}\cdot\mathtt{x})$ *(if $\mathtt{x}$ is not free in $\mathtt{t}$)*

Proof. Some derivations can be found within Figures 3 and 4. For *D*1 and *D*2 assume $\mathtt{y}$ is not free in $\mathtt{t}$, for *D*4 assume $\mathtt{x}$ is not free in $\mathtt{t}$.

$\nvdash \mathtt{x} :\!- \mathtt{y}$ and $\nvdash \mathtt{x}\rightsquigarrow\mathtt{y}$ follows from the syntax directedness of the derivation sequent rules.[2] □

We now prove a theorem relating the term sequent treatment of λ-calculus to familiar treatments of the λ-calculus in terms of $\alpha\beta\eta$-conversion. First we need to associate a λ-term to every tree.

Definition 3.3 If Θ is a tree then $\mathit{trm}(\Theta)$ is a term defined inductively by $\mathit{trm}(\mathtt{t})\equiv\mathtt{t}$ and $\mathit{trm}(\langle\Theta_1,\Theta_2\rangle) \equiv \mathit{trm}(\Theta_1)\cdot\mathit{trm}(\Theta_2)$.

Lemma 3.4 $\Gamma \vdash \Theta :\!- \mathit{trm}(\Theta)$

Proof. The proof is by an easy induction on Θ. □

Definition 3.5 If $\mathtt{t}_1$ and $\mathtt{t}_2$ are λ-terms then let $\mathtt{t}_1 \twoheadrightarrow_{\alpha\beta\eta} \mathtt{t}_2$ mean that $\mathtt{t}_1$ can be rewritten to $\mathtt{t}_2$ by means of α-conversion, β-reduction and η-expansion.

Theorem 3.6 $\vdash \Theta :\!- \mathtt{t}$ *if and only if* $\mathit{trm}(\Theta) \twoheadrightarrow_{\alpha\beta\eta} \mathtt{t}$.

Proof. The 'if' direction follows by Theorem 3.2 (right column), (Cut_λ) and Lemma 3.4.

The 'only if' direction follows by induction on derivations. For example, suppose the derivation ends with (λR): $\dfrac{\overset{\Pi}{\Gamma \vdash \langle\Theta,\mathtt{x}\rangle :\!- \mathtt{t}}}{\Gamma \vdash \Theta :\!- \lambda\mathtt{x}.\mathtt{t}}\,(\lambda R)$. Then by the induction hypothesis on Π and Definition 3.3 $\mathit{trm}(\Theta)\cdot\mathtt{x} \twoheadrightarrow_{\alpha\beta\eta} \mathtt{t}$, but then by congruence $\lambda\mathtt{x}.(\mathit{trm}(\Theta)\cdot\mathtt{x}) \twoheadrightarrow_{\alpha\beta\eta} \lambda\mathtt{x}.\mathtt{t}$. Thus by η-expansion (and the transitivity of $\twoheadrightarrow_{\alpha\beta\eta}$) we have that $\mathit{trm}(\Theta) \twoheadrightarrow_{\alpha\beta\eta} \lambda\mathtt{x}.\mathtt{t}$. □

In Figure 4 we present some complex derivations. We do not necessarily decompose λ-terms to normal forms (the λ-calculus is untyped; there may not be a normal form). Derivation *D*5 exemplifies how term sequents can handle a λ-terms that does not reduce to a normal form. Derivation *D*8 exemplifies that term sequents are no rewrite system for the λ-calculus in disguise; we derive that a reduction holds *if* $\mathtt{y}$ is reducible to $\mathtt{z}$. As discussed in the Introduction, a rewrite system cannot hypothesise rewrites.

4 Interreducivity and intersubstitutivity

Definition 4.1 Write $\mathtt{t} \leftrightsquigarrow \mathtt{t}'$ for $\mathtt{t}\rightsquigarrow\mathtt{t}' \wedge \mathtt{t}'\rightsquigarrow\mathtt{t}$.

[2] Actually, the rules are not absolutely syntax directed as Derivation *D*5 shows. It would be surprising if they were for there is no general normal form for untyped λ-terms. However, it is clear that no derivation rules can reduce the overall complexity of a term sequent to containing only atomic terms.

$$(D5)\quad \dfrac{\begin{array}{c}\vdots\ \text{Theorem 3.1}\\ \vdash \mathtt{y}\cdot[\lambda\mathtt{x}.(\mathtt{y}\cdot(\mathtt{x}\cdot\mathtt{x}))\cdot\lambda\mathtt{x}.(\mathtt{y}\cdot(\mathtt{x}\cdot\mathtt{x}))] \mathrel{:-} \mathtt{y}\cdot[\lambda\mathtt{x}.(\mathtt{y}\cdot(\mathtt{x}\cdot\mathtt{x}))\cdot\lambda\mathtt{x}.(\mathtt{y}\cdot(\mathtt{x}\cdot\mathtt{x}))]\end{array}}{\vdash \langle\lambda\mathtt{x}.(\mathtt{y}\cdot(\mathtt{x}\cdot\mathtt{x})),\lambda\mathtt{x}.(\mathtt{y}\cdot(\mathtt{x}\cdot\mathtt{x}))\rangle \mathrel{:-} \mathtt{y}\cdot[\lambda\mathtt{x}.(\mathtt{y}\cdot(\mathtt{x}\cdot\mathtt{x}))\cdot\lambda\mathtt{x}.(\mathtt{y}\cdot(\mathtt{x}\cdot\mathtt{x}))]}\,{\scriptstyle(\lambda L)}$$

$$(D6)\quad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \mathtt{y}\mathrel{:-}\mathtt{y}}\,{\scriptstyle(Ax_{:-})}}{\mathtt{x}\rightsquigarrow\mathtt{y}\vdash\mathtt{x}\mathrel{:-}\mathtt{y}}\,{\scriptstyle(\rightsquigarrow L)}}{\mathtt{x}\rightsquigarrow\mathtt{y},\mathtt{y}\rightsquigarrow\mathtt{z}\vdash\mathtt{x}\mathrel{:-}\mathtt{z}}\,{\scriptstyle(\rightsquigarrow L)}}{\mathtt{x}\rightsquigarrow\mathtt{y},\mathtt{y}\rightsquigarrow\mathtt{z}\vdash\mathtt{x}\rightsquigarrow\mathtt{z}}\,{\scriptstyle(\rightsquigarrow R)}
\qquad
(D7)\quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \mathtt{z}\mathrel{:-}\mathtt{z}}\,{\scriptstyle(Ax_{:-})}}{\vdash\langle\lambda\mathtt{x}.\mathtt{x},\mathtt{z}\rangle\mathrel{:-}\mathtt{z}}\,{\scriptstyle(\lambda L)}}{\mathtt{y}\rightsquigarrow\lambda\mathtt{x}.\mathtt{x}\vdash\langle\mathtt{y},\mathtt{z}\rangle\mathrel{:-}\mathtt{z}}\,{\scriptstyle(\rightsquigarrow L)}}{\mathtt{y}\rightsquigarrow\lambda\mathtt{x}.\mathtt{x}\vdash\mathtt{y}\cdot\mathtt{z}\mathrel{:-}\mathtt{z}}\,{\scriptstyle(\cdot R)}}{\mathtt{y}\rightsquigarrow\lambda\mathtt{x}.\mathtt{x}\vdash(\mathtt{y}\cdot\mathtt{z})\rightsquigarrow\mathtt{z}}\,{\scriptstyle(\rightsquigarrow R)}$$

$$(D8)\quad \dfrac{\dfrac{\dfrac{}{\mathtt{y}\rightsquigarrow\mathtt{z}\vdash\mathtt{y}\mathrel{:-}\mathtt{y}}\,{\scriptstyle(Ax_{:-})}\qquad \dfrac{\dfrac{\dfrac{}{\vdash\mathtt{y}\mathrel{:-}\mathtt{y}}\,{\scriptstyle(Ax_{:-})}\qquad \dfrac{\dfrac{}{\vdash\mathtt{z}\mathrel{:-}\mathtt{z}}\,{\scriptstyle(Ax_{:-})}\quad\dfrac{}{\vdash\mathtt{y}\mathrel{:-}\mathtt{y}}\,{\scriptstyle(Ax_{:-})}}{\vdash\langle\mathtt{z},\mathtt{y}\rangle\mathrel{:-}\mathtt{z}\cdot\mathtt{y}}\,{\scriptstyle(\cdot R)}}{\mathtt{y}\rightsquigarrow\mathtt{z}\vdash\langle\mathtt{y},\mathtt{y}\rangle\mathrel{:-}\mathtt{z}\cdot\mathtt{y}}\,{\scriptstyle(\rightsquigarrow L)}}{\mathtt{y}\rightsquigarrow\mathtt{z}\vdash\mathtt{y}\cdot\mathtt{y}\mathrel{:-}\mathtt{z}\cdot\mathtt{y}}\,{\scriptstyle(\cdot L)}}{\dfrac{\mathtt{y}\rightsquigarrow\mathtt{z}\vdash\langle\lambda\mathtt{x}.(\mathtt{x}\cdot\mathtt{x}),\mathtt{y}\rangle\mathrel{:-}\mathtt{z}\cdot\mathtt{y}}{\mathtt{y}\rightsquigarrow\mathtt{z}\vdash\lambda\mathtt{x}.(\mathtt{x}\cdot\mathtt{x})\cdot\mathtt{y}\mathrel{:-}\mathtt{z}\cdot\mathtt{y}}\,{\scriptstyle(\cdot L)}}\,{\scriptstyle(\lambda L)}$$

Fig. 4. Example derivations in term sequent λ-calculus

In the light of Definition 4.1 and the rules in Figure 1 we might look at Theorem 3.6 and jump to the conclusion that $\leftrightsquigarrow$ is just α-equivalence — but this conclusion is false. In $\Gamma \vdash \Theta \mathrel{:-} \mathtt{t}$ and $\Gamma \vdash \Delta$ we can assume reductions in Γ; so $\leftrightsquigarrow$ holds or fails to hold in the context of some assumptions, which can assert reductions that are not α-equivalences.

We will show that $\dfrac{\Gamma\vdash \mathtt{A}[\mathtt{x}/\mathtt{t}],\ \Delta}{\Gamma,\ \mathtt{t}\leftrightsquigarrow\mathtt{t}'\vdash\mathtt{A}[\mathtt{x}/\mathtt{t}'],\ \Delta}$ is admissible (Theorem 4.3).

Lemma 4.2 *–* $\Gamma\vdash\mathtt{t}_2\mathrel{:-}\mathtt{t}_1$ *and* $\Gamma\vdash\langle\ldots\mathtt{s}[\mathtt{x}/\mathtt{t}_1]\ldots\rangle\mathrel{:-}\mathtt{s}'$ *imply* $\Gamma\vdash\langle\ldots\mathtt{s}[\mathtt{x}/\mathtt{t}_2]\ldots\rangle\mathrel{:-}\mathtt{s}'$.
– $\Gamma\vdash\mathtt{t}_1\mathrel{:-}\mathtt{t}_2$ *and* $\Gamma\vdash\Theta\mathrel{:-}\mathtt{s}[\mathtt{x}/\mathtt{t}_1]$ *imply* $\Gamma\vdash\Theta\mathrel{:-}\mathtt{s}[\mathtt{x}/\mathtt{t}_2]$.

Proof. The first part is by induction on the derivation of $\Gamma\vdash\langle\ldots\mathtt{s}[\mathtt{x}/\mathtt{t}_1]\ldots\rangle\mathrel{:-}\mathtt{s}'$. There are a number of cases for the final rule of the derivation. We consider some of them here:

- The case $(Ax_{:-})$. Then $\mathtt{s}\equiv\mathtt{x}$ or, for some $\mathtt{y}\not\equiv\mathtt{x}$, $\mathtt{s}\equiv\mathtt{y}$. In the the first case the result follows from the assumption that $\Gamma\vdash\mathtt{t}_2\mathrel{:-}\mathtt{t}_1$ and (Cut_λ), otherwise $\mathtt{s}[\mathtt{x}/\mathtt{t}_1]\equiv\mathtt{y}[\mathtt{x}/\mathtt{t}_1]\equiv\mathtt{y}[\mathtt{x}/\mathtt{t}_2]\equiv\mathtt{s}$.

  We may now assume that $\mathtt{s}$ is not atomic. For otherwise, regardless of the final rule application, $\mathtt{s}\equiv\mathtt{x}$ or $\mathtt{s}\equiv\mathtt{y}$ for some $\mathtt{y}\not\equiv\mathtt{x}$ and the result follows as above.
- The cases where last rule applies to some term other than $\mathtt{s}[\mathtt{x}/\mathtt{t}_1]$. Then the result follows easily by the induction hypothesis on the shorter derivation of the premise.
- The cases where the last rule applies to $\mathtt{s}[\mathtt{x}/\mathtt{t}_1]$. Then the result follows again by the induction hypothesis. For example suppose $\mathtt{s}\equiv\lambda\mathtt{y}.\mathtt{r}$ and is derived by (λL):

$$\dfrac{\begin{array}{c}\Pi_1\\ \Gamma\vdash\Theta\mathrel{:-}\mathtt{r}'\end{array}\quad \begin{array}{c}\Pi_2\\ \Gamma\vdash\langle\ldots\mathtt{r}[\mathtt{x}/\mathtt{t}_1][\mathtt{y}/\mathtt{r}']\ldots\rangle\mathrel{:-}\mathtt{t}\end{array}}{\Gamma\vdash\langle\ldots\langle\lambda\mathtt{y}.\mathtt{r}[\mathtt{x}/\mathtt{t}_1],\Theta\rangle\ldots\rangle\mathrel{:-}\mathtt{t}}\,{\scriptstyle(\lambda L)}$$

  We may suppose that $\mathtt{x}$ is not free in $\mathtt{r}'$ and $\mathtt{y}$ is not free in $\mathtt{t}_1$ and so

$\mathtt{r}[\mathtt{x}/\mathtt{t}_1][\mathtt{y}/\mathtt{r}'] \equiv \mathtt{r}[\mathtt{y}/\mathtt{r}'][\mathtt{x}/\mathtt{t}_1]$. Thus $\Gamma \vdash \langle \dots \mathtt{r}[\mathtt{y}/\mathtt{r}'][\mathtt{x}/\mathtt{t}_1] \dots \rangle \coloneq \mathtt{t}$ is the conclusion of Π_2. So there is a derivation Π_2' of $\Gamma \vdash \langle \dots \mathtt{r}[\mathtt{y}/\mathtt{r}'][\mathtt{x}/\mathtt{t}_2] \dots \rangle \coloneq \mathtt{t}$ by induction hypothesis. So we have that:

$$\dfrac{\begin{array}{c}\Pi_1\\ \Gamma \vdash \Theta \coloneq \mathtt{r}'\end{array} \quad \begin{array}{c}\Pi_2'\\ \Gamma \vdash \langle \dots \mathtt{r}[\mathtt{y}/\mathtt{r}'][\mathtt{x}/\mathtt{t}_2] \dots \rangle \coloneq \mathtt{t}\end{array}}{\Gamma \vdash \langle \dots \langle \lambda \mathtt{y}.\mathtt{r}[\mathtt{x}/\mathtt{t}_2], \Theta \rangle \dots \rangle \coloneq \mathtt{t}}\ (\lambda L)$$

since $\mathtt{r}[\mathtt{y}/\mathtt{r}'][\mathtt{x}/\mathtt{t}_1] \equiv \mathtt{r}[\mathtt{x}/\mathtt{t}_1][\mathtt{y}/\mathtt{r}']$. The remaining possibilities involve little more complexity.

The second part is proved similarly to the first part, by induction on the derivation of $\Gamma \vdash \Theta \coloneq \mathtt{s}[\mathtt{x}/\mathtt{t}_1]$.

- The base case, as with the first part, is when $\Gamma \vdash \Theta \coloneq \mathtt{s}[\mathtt{x}/\mathtt{t}_1]$ is derived by $(Ax_{\coloneq})$ or when $\mathtt{s}$ is atomic. In each subcase the result follows easily by (Cut_λ). For example, if $\mathtt{s} \not\equiv \mathtt{x}$ then we have: $\dfrac{\Gamma \vdash \Theta \coloneq \mathtt{s}[\mathtt{x}/\mathtt{t}_1] \quad \Gamma \vdash \mathtt{t}_1 \coloneq \mathtt{t}_2}{\Gamma \vdash \Theta \coloneq \mathtt{t}_2}\ (Cut_\lambda)$.
- The remaining cases are uncomplicated.

$\square$

Theorem 4.3 – $\Gamma \vdash \mathtt{t}_1 \coloneq \mathtt{t}_2$, $\Gamma \vdash \mathtt{t}_2 \coloneq \mathtt{t}_1$ *and* $\Gamma \vdash \mathtt{A}[\mathtt{x}/\mathtt{t}_1], \Delta$ *imply* $\Gamma \vdash \mathtt{A}[\mathtt{x}/\mathtt{t}_2], \Delta$.
– $\Gamma \vdash \mathtt{A}[\mathtt{x}/\mathtt{t}_1], \Delta$ *implies* $\Gamma, \mathtt{t}_1 \leftrightsquigarrow \mathtt{t}_2 \vdash \mathtt{A}[\mathtt{x}/\mathtt{t}_2], \Delta$.

Proof. The first part follows by induction on the derivation of $\Gamma \vdash \mathtt{A}[\mathtt{x}/\mathtt{t}_1], \Delta$. The base case is where $\mathtt{A} = \mathtt{s}_1 \rightsquigarrow \mathtt{s}_2$ and the derivation ends with $(\rightsquigarrow R)$ applied to the premise $\Gamma \vdash \mathtt{t}_1[\mathtt{x}/\mathtt{t}_1] \coloneq \mathtt{t}_2[\mathtt{x}/\mathtt{t}_1]$. The result then follows by Lemma 4.2. The inductive cases are straightforward.

The second part follows from the first part, (Cut_λ) and from the fact that $\Gamma, \mathtt{t}_1 \rightsquigarrow \mathtt{t}_2, \mathtt{t}_2 \rightsquigarrow \mathtt{t}_1 \vdash \mathtt{t}_1 \coloneq \mathtt{t}_2$ and $\Gamma, \mathtt{t}_1 \rightsquigarrow \mathtt{t}_2, \mathtt{t}_2 \rightsquigarrow \mathtt{t}_1 \vdash \mathtt{t}_2 \coloneq \mathtt{t}_1$. $\square$

So $\mathtt{t}_1 \leftrightsquigarrow \mathtt{t}_2$ functions like a substitutional equality on sentences. See Section 7 for a (weaker) notion of equality corresponding with $\alpha\beta\eta$-equality in a suitable formal sense.

5 Cut elimination

Cut elimination is proved in two stages: eliminate the term sequent rule (Cut_λ), then eliminate the sentence sequent rule (Cut).

Eliminating (Cut_λ) is somewhat simplified compared to the propositional case, because term sequents lack structural rules like contraction and weakening. First, some definitions.

5.1 Height and grade of term sequents

Definition 5.1 Define the ***height*** of a term sequent in a derivation by:

– A term sequent of the form $\Gamma \vdash \mathtt{t} \coloneq \mathtt{t}$ has height 0.

– The height of any other term sequent (in a derivation) is the sum of the heights of its premises plus 1.

Call the height of a derivation the height of its final sequent.

The height of a term sequent, therefore, is the size of the derivation tree extending back as far as the last instances of $\Gamma \vdash \mathtt{t} :\!- \mathtt{t}$. We can define something equivalent for sentence sequents.

Definition 5.2 The ***grade*** of a term is the number of occurrences in it of the symbols $\cdot$ and λ. For example, $(\lambda\mathtt{x}.(\lambda\mathtt{y}.\mathtt{x}\cdot\mathtt{y}))\cdot\mathtt{z}$ has grade 4.

Definition 5.3 For an instance of (Cut_λ) $\dfrac{\Gamma \vdash \Theta :\!- \mathtt{s} \quad \Gamma \vdash \langle \ldots \mathtt{s} \ldots \rangle :\!- \mathtt{t}}{\Gamma \vdash \langle \ldots \Theta \ldots \rangle :\!- \mathtt{t}}$ call $\mathtt{s}$ the ***cut term***, call the ***grade*** of the instance the grade of the cut sentence. Call the ***height*** of the instance the height of its conclusion.

5.2 Rank and degree of sentence sequents

Definition 5.4 Define the ***rank*** of a *sentence sequent* in a derivation by:

– The conclusion of $(T_\bot)$ or $(\leadsto R)$ has rank 0.

– The rank of any other term sequent (in a derivation) is the sum of the ranks of its premises plus 1.

The rank of a derivation is the rank of its conclusion.

Definition 5.5 The ***degree*** of a sentence is the number of occurrences in it of the symbols $\wedge, \neg, \forall$. For example, $\forall\mathtt{x}.\neg\forall\mathtt{y}.\exists\mathtt{z}.(\mathtt{x}\leadsto\mathtt{y} \wedge \mathtt{x}\leadsto\mathtt{z})$ has degree 5.

Definition 5.6 For an instance of (Cut) $\dfrac{\Gamma \vdash \mathtt{C}, \Delta \quad \Gamma, \mathtt{C} \vdash \Delta}{\Gamma \vdash \Delta}$ call $\mathtt{C}$ the ***cut sentence***, call the ***degree*** of the instance the degree of the cut sentence. Call the ***rank*** of the instance the rank of its conclusion.

5.3 Cut_λ-elimination

We must first prove a lemma on the uniform substitution of variables for terms.

Lemma 5.7 – $\Gamma \vdash \Theta :\!- \mathtt{t}$ *implies* $\Gamma[\mathtt{x}/\mathtt{t}'] \vdash \Theta[\mathtt{x}/\mathtt{t}'] :\!- \mathtt{t}[\mathtt{x}/\mathtt{t}']$ *with no greater height.*

– *If* $\Gamma \vdash \Delta$ *then* $\Gamma[\mathtt{x}/\mathtt{t}] \vdash \Delta[\mathtt{x}/\mathtt{t}], \Delta'[\mathtt{x}/\mathtt{t}]$ *with the same rank.*

Proof. Both parts follow by induction on the derivation. □

Lemma 5.8 *If* $\Gamma \vdash \Theta :\!- \mathtt{t}$ *then* $\Gamma, \mathtt{A} \vdash \Theta :\!- \mathtt{t}$ *with no more instances of* (Cut_λ).

Proof. By a simple induction on the derivation. □

Theorem 5.9 – *If* $\Gamma \vdash \Theta :\!- \mathtt{t}$ *is derivable using exactly one instance of* (Cut_λ) *then it is derivable, with no greater height, using none.*

– *If* $\Gamma \vdash \Theta :\!- \mathtt{t}$ *is derivable then it is derivable without* (Cut_λ).

Proof.

– The first part is proved by induction on the pair (g, h), lexicographically ordered, where g is the height and h is the grade of the (Cut_λ). As with familiar cut elimination theorems, we permute cuts and eliminate essential cases. The essential cases and examples of permutation rules are given in Figures 5 and 6, the base case is given by Lemma 3.1.

$$\cfrac{\cfrac{}{\Gamma\vdash \mathtt{t}_1 \mathrel{:\!-} \mathtt{t}_1}{\scriptstyle (Ax_{:-})} \qquad \cfrac{\Pi}{\Gamma\vdash \langle\ldots\mathtt{t}_1\ldots\rangle \mathrel{:\!-} \mathtt{t}_2}}{\Gamma\vdash \langle\ldots\mathtt{t}_1\ldots\rangle \mathrel{:\!-} \mathtt{t}_2}{\scriptstyle (Cut_\lambda)} \quad\longmapsto\quad \cfrac{\Pi}{\Gamma\vdash \langle\ldots\mathtt{t}_1\ldots\rangle \mathrel{:\!-} \mathtt{t}_2}$$

$$\cfrac{\cfrac{\Pi}{\Gamma\vdash \langle\ldots\Theta\ldots\rangle \mathrel{:\!-} \mathtt{t}} \qquad \cfrac{}{\Gamma\vdash \mathtt{t} \mathrel{:\!-} \mathtt{t}}{\scriptstyle (Ax_{:-})}}{\Gamma\vdash \langle\ldots\Theta\ldots\rangle \mathrel{:\!-} \mathtt{t}}{\scriptstyle (Cut_\lambda)} \quad\longmapsto\quad \cfrac{\Pi}{\Gamma\vdash \langle\ldots\Theta\ldots\rangle \mathrel{:\!-} \mathtt{t}}$$

$$\cfrac{\cfrac{\cfrac{\Pi_1}{\Gamma\vdash\langle\Theta_1,\mathtt{x}\rangle \mathrel{:\!-} \mathtt{t}_1}}{\Gamma\vdash\Theta_1 \mathrel{:\!-} \lambda\mathtt{x}.\mathtt{t}_1}{\scriptstyle (\lambda R)} \qquad \cfrac{\cfrac{\Pi_2}{\Gamma\vdash\Theta_2 \mathrel{:\!-} \mathtt{t}_2} \quad \cfrac{\Pi_3}{\Gamma\vdash\langle\ldots\mathtt{t}_1[\mathtt{x}/\mathtt{t}]\ldots\rangle \mathrel{:\!-} \mathtt{t}_2}}{\Gamma\vdash\langle\ldots\langle\lambda\mathtt{x}.\mathtt{t}_1,\Theta_2\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}}{\scriptstyle (\lambda L)}}{\Gamma\vdash\langle\ldots\langle\Theta_1,\Theta_2\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}_2}{\scriptstyle (Cut_\lambda)}$$

$$\longmapsto\quad \cfrac{\cfrac{\Pi_2}{\Gamma\vdash\Theta_2 \mathrel{:\!-} \mathtt{t}_2} \qquad \cfrac{\cfrac{\Pi_1[\mathtt{x}/\mathtt{t}]}{\Gamma\vdash\langle\Theta_1,\mathtt{t}\rangle \mathrel{:\!-} \mathtt{t}_1[\mathtt{x}/\mathtt{t}]} \quad \cfrac{\Pi_3}{\Gamma\vdash\langle\ldots\mathtt{t}_1[\mathtt{x}/\mathtt{t}]\ldots\rangle \mathrel{:\!-} \mathtt{t}_2}}{\Gamma\vdash\langle\ldots\langle\Theta_1,\mathtt{t}\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}}{\scriptstyle (Cut_\lambda)}}{\Gamma\vdash\langle\ldots\langle\Theta_1,\Theta_2\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}_2}{\scriptstyle (Cut_\lambda)}$$

where $\Pi_1[\mathtt{x}/\mathtt{t}]$ results from Π_1 by replacing $\mathtt{x}$ uniformly in it by $\mathtt{t}$ (see Lemma 5.7).

$$\cfrac{\cfrac{\cfrac{\Pi_1}{\Gamma\vdash\Theta_1 \mathrel{:\!-} \mathtt{t}_1} \quad \cfrac{\Pi_2}{\Gamma\vdash\Theta_2 \mathrel{:\!-} \mathtt{t}_2}}{\Gamma\vdash\langle\Theta_1,\Theta_2\rangle \mathrel{:\!-} \mathtt{t}_1{\cdot}\mathtt{t}_2}{\scriptstyle (\cdot R)} \qquad \cfrac{\cfrac{\Pi_3}{\Gamma\vdash\langle\ldots\langle\mathtt{t}_1,\mathtt{t}_2\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}}}{\Gamma\vdash\langle\ldots\langle\mathtt{t}_1{\cdot}\mathtt{t}_2\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}}{\scriptstyle (\cdot L)}}{\Gamma\vdash\langle\ldots\langle\Theta_1,\Theta_2\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}}{\scriptstyle (Cut_\lambda)}$$

$$\longmapsto\quad \cfrac{\cfrac{\Pi_1}{\Gamma\vdash\Theta_1 \mathrel{:\!-} \mathtt{t}_1} \qquad \cfrac{\cfrac{\Pi_2}{\Gamma\vdash\Theta_2 \mathrel{:\!-} \mathtt{t}_2} \quad \cfrac{\Pi_3}{\Gamma\vdash\langle\ldots\langle\mathtt{t}_1,\mathtt{t}_2\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}}}{\Gamma\vdash\langle\ldots\langle\mathtt{t}_1,\Theta_2\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}}{\scriptstyle (Cut_\lambda)}}{\Gamma\vdash\langle\ldots\langle\Theta_1,\Theta_2\rangle\ldots\rangle \mathrel{:\!-} \mathtt{t}_t}{\scriptstyle (Cut_\lambda)}$$

Fig. 5. Some essential cases for Cut_λ-elimination

In all cases we replace an instance of (Cut_λ) with a number of new instances with lesser heights or degrees, these may be eliminated by the induction hypothesis. The result as a whole is made easier by the fact the the derivation system contains no 'exponential' structural rules such as weakening or contraction (exponential with regard to their effect on derivation complexity).

– The second part follows from the first by induction on the structure of the derivation.

□

We must prove that term sequents interact well with sentence sequents.

Theorem 5.10 $\Gamma\vdash \mathtt{t}_1 \mathrel{:\!-} \mathtt{t}_2$ *and* $\Gamma, \mathtt{t}_1{\leadsto}\mathtt{t}_2 \vdash \Theta \mathrel{:\!-} \mathtt{t}$ *imply* $\Gamma\vdash\Theta \mathrel{:\!-} \mathtt{t}$

Proof. By induction on the height of the derivation that $\Gamma, \mathtt{t}_1{\leadsto}\mathtt{t}_2 \vdash \Theta \mathrel{:\!-} \mathtt{t}$. If the final rule application is $({\leadsto}L)$ introducing $\mathtt{t}_1{\leadsto}\mathtt{t}_2$ then the result follows by (Cut_λ) and then Theorem 5.9. [3] □

Full cut elimination now follows:

Theorem 5.11 – *If* $\Gamma\vdash\Delta$ *is derivable using exactly one instance of* (Cut) *then it is derivable using none.*

– *If* $\Gamma\vdash\Delta$ *is derivable then it is derivable without using* (Cut).

[3] Since (Cut) is a rule applying only to sentence sequents it cannot feature in the derivations of $\Gamma\vdash \mathtt{t}_1 \mathrel{:\!-} \mathtt{t}_2$ and $\Gamma, \mathtt{t}_1{\leadsto}\mathtt{t}_2 \vdash \Theta \mathrel{:\!-} \mathtt{t}$.

$$\dfrac{\dfrac{\overset{\Pi_1}{\Gamma \vdash \Theta_1 \mathrel{:-} \mathtt{t}_1} \quad \overset{\Pi_2}{\Gamma \vdash \langle \ldots \mathtt{t}_2 \ldots \rangle \mathrel{:-} \mathtt{t}}}{\Gamma' \vdash \langle \ldots \Theta_1 \ldots \rangle \mathrel{:-} \mathtt{t}}\,{\scriptstyle (\leadsto L)} \quad \overset{\Pi_3}{\Gamma' \vdash \langle \ldots \mathtt{t} \ldots \rangle \mathrel{:-} \mathtt{t}_3}}{\Gamma, \mathtt{t}_1 \leadsto \mathtt{t}_2 \vdash \langle \ldots \langle \ldots \Theta_1 \ldots \rangle \ldots \rangle \mathrel{:-} \mathtt{t}_3}\,{\scriptstyle (Cut_\lambda)}$$

$$\longmapsto \quad \dfrac{\overset{\Pi_1'}{\Gamma' \vdash \Theta_1 \mathrel{:-} \mathtt{t}_1} \quad \dfrac{\overset{\Pi_2'}{\Gamma' \vdash \langle \ldots \mathtt{t}_2 \ldots \rangle \mathrel{:-} \mathtt{t}} \quad \overset{\Pi_3}{\Gamma, \mathtt{t}_1 \leadsto \mathtt{t}_2 \vdash \langle \ldots \mathtt{t} \ldots \rangle \mathrel{:-} \mathtt{t}_3}}{\Gamma' \vdash \langle \ldots \langle \ldots \mathtt{t}_2 \ldots \rangle \ldots \rangle \mathrel{:-} \mathtt{t}_3}\,{\scriptstyle (Cut_\lambda)}}{\Gamma' \vdash \langle \ldots \langle \ldots \Theta_1 \ldots \rangle \ldots \rangle \mathrel{:-} \mathtt{t}_3}\,{\scriptstyle (\leadsto L)}$$

where Γ' is $\Gamma, \mathtt{t}_1 \leadsto \mathtt{t}_2$, and Π_1', Π_2' are obtained from Π_1, Π_2 by weakening with $\mathtt{t}_1 \approx \mathtt{t}_2$ (see Lemma 5.8)

$$\dfrac{\overset{\Pi_1}{\Gamma \vdash \Theta \mathrel{:-} \mathtt{t}} \quad \dfrac{\overset{\Pi_2}{\Gamma \vdash \langle \ldots \mathtt{t} \ldots \rangle \mathrel{:-} \mathtt{t}_1} \quad \overset{\Pi_3}{\Gamma \vdash \langle \ldots \mathtt{t}_2[\mathtt{x}/\mathtt{t}_1] \ldots \rangle \mathrel{:-} \mathtt{t}_3}}{\Gamma \vdash \langle \ldots \langle \lambda \mathtt{x}.\mathtt{t}_2, \langle \ldots \mathtt{t} \ldots \rangle \rangle \ldots \rangle \mathrel{:-} \mathtt{t}_3}\,{\scriptstyle (\lambda L)}}{\Gamma \vdash \langle \ldots \langle \lambda \mathtt{x}.\mathtt{t}_2, \langle \ldots \Theta \ldots \rangle \rangle \ldots \rangle \mathrel{:-} \mathtt{t}_3}\,{\scriptstyle (Cut_\lambda)}$$

$$\longmapsto \quad \dfrac{\dfrac{\overset{\Pi}{\Gamma \vdash \Theta \mathrel{:-} \mathtt{t}} \quad \overset{\Pi_2}{\Gamma \vdash \langle \ldots \mathtt{t} \ldots \rangle \mathrel{:-} \mathtt{t}_1}}{\Gamma \vdash \langle \ldots \Theta_1 \ldots \rangle \mathrel{:-} \mathtt{t}_1}\,{\scriptstyle (Cut_\lambda)} \quad \overset{\Pi_3}{\Gamma \vdash \langle \ldots \mathtt{t}_2[\mathtt{x}/\mathtt{t}_1] \ldots \rangle \mathrel{:-} \mathtt{t}_3}}{\Gamma \vdash \langle \ldots \langle \lambda \mathtt{x}.\mathtt{t}_2, \langle \ldots \Theta \ldots \rangle \rangle \ldots \rangle \mathrel{:-} \mathtt{t}_3}\,{\scriptstyle (\lambda L)}$$

Fig. 6. Some permutation cases for Cut_λ-elimination

Proof.

– The first part is by induction on the pair (d, r), lexicographically ordered, where d is the degree and r is the rank of the (Cut). The base case is where the cut formula is of the form $\mathtt{t}_1 \leadsto \mathtt{t}_2$:

$$\dfrac{\dfrac{\overset{\Pi_1}{\Gamma \vdash \mathtt{t}_1 \mathrel{:-} \mathtt{t}_2}}{\Gamma \vdash \mathtt{t}_1 \leadsto \mathtt{t}_2, \Delta}\,{\scriptstyle (\leadsto R)} \quad \dfrac{\overset{\Pi_2}{\Gamma, \mathtt{t}_1 \leadsto \mathtt{t}_2 \vdash \mathtt{t} \mathrel{:-} \mathtt{t}'}}{\Gamma, \mathtt{t}_1 \leadsto \mathtt{t}_2 \vdash \mathtt{t} \leadsto \mathtt{t}', \Delta}\,{\scriptstyle (\leadsto R)}}{\Gamma \vdash \Delta}\,{\scriptstyle (Cut)} \quad \longmapsto \quad \dfrac{\begin{array}{c}\Pi_1 \quad \Pi_2 \\ \vdots \ \text{Theorem 5.10} \\ \Gamma \vdash \mathtt{t} \mathrel{:-} \mathtt{t}'\end{array}}{\Gamma \vdash \mathtt{t} \leadsto \mathtt{t}', \Delta}\,{\scriptstyle (\leadsto R)}$$

The remaining cases are as for cut elimination on first order logic.

– The second part follows by induction on the derivation.

□

6 Uniform derivations

Definition 6.1 A derivation of a term sequent is ***uniform*** if it is (Cut_λ) free and for every term sequent $\Gamma \vdash \Theta \mathrel{:-} \mathtt{t}$ in it, if $\mathtt{t}$ is not atomic then that sequent is the conclusion of an instance of a rule applying to $\mathtt{t}$ (i.e. (λR) or $(\cdot R)$).

It is of interest to determine which term sequents are derivable by uniform derivations. To guarantee that derivations are uniform we need to modify the term sequent rules, in particular the left rules, and we must tweak our definition of a tree:

Definition 6.2 – *Redefine **trees** by:* $\Theta ::= \mathtt{t} \mid \langle \Theta_1, \Theta_2 \rangle \mid \langle \Theta_1; \Theta_2 \rangle$.

The difference between $\langle \Theta_1, \Theta_2 \rangle$ and $\langle \Theta_1; \Theta_2 \rangle$ is purely for bookkeeping, see Remark 6.5.

Definition 6.3 – The modified rule $(\rightsquigarrow L^-)$ is rule $(\rightsquigarrow L)$

$$\frac{\Gamma \vdash \Theta :- \mathtt{t}_1 \quad \Gamma \vdash \langle \ldots \mathtt{t}_2 \ldots \rangle :- \mathtt{t}}{\Gamma, \mathtt{t}_1 \rightsquigarrow \mathtt{t}_2 \vdash \langle \ldots \Theta \ldots \rangle :- \mathtt{t}}$$

with an additional restriction that the free variables of (terms appearing in) Θ be free also in $\mathtt{t}_2$ (intuitively, $fv(\Theta) \subseteq fv(\mathtt{t}_2)$).
– The modified rule (λL^-) is

$$\frac{\Gamma \vdash \Theta :- \mathtt{t}_1 \quad \Gamma \vdash \langle \ldots \mathtt{t}_2[\mathtt{x}/\mathtt{t}_1] \ldots \rangle :- \mathtt{t}}{\Gamma \vdash \langle \ldots \langle \lambda \mathtt{x}.\mathtt{t}_2; \Theta \rangle \ldots \rangle :- \mathtt{t}} \ (\lambda L^-)$$

(note the semicolon in the conclusion) with the restriction that the free variables of Θ be free also in $\mathtt{t}_2[\mathtt{x}/\mathtt{t}_1]$.
– The modified rule $(\cdot L^-)$ is

$$\frac{\Gamma \vdash \langle \ldots \langle \mathtt{t}_1; \mathtt{t}_2 \rangle \ldots \rangle :- \mathtt{t}}{\Gamma \vdash \langle \ldots \mathtt{t}_1 \cdot \mathtt{t}_2 \ldots \rangle :- \mathtt{t}} \ (\cdot L^-)$$

(note the semicolon in the premise).
– We add an extra (λR) rule for the case of a semicolon:

$$\frac{\Gamma \vdash \langle \Theta; \mathtt{x} \rangle :- \mathtt{t}}{\Gamma \vdash \Theta :- \lambda \mathtt{x}.\mathtt{t}} \ (\lambda R)$$

(with the restriction that $\mathtt{x}$ is not free in Θ or Γ).
– The other rules are unchanged. In particular $(\cdot R)$ is *not* modified so as to introduce $\langle \Theta_1; \Theta_2 \rangle$ on the left, it introduces only $\langle \Theta_1, \Theta_2 \rangle$.

Theorem 6.4 *If* $\Gamma \vdash \Theta :- \mathtt{t}$ *is derivable using the modified term sequent rules of Definition 6.3, then it is derivable by a uniform derivation.*

Proof. By Theorem 5.9 (it remains valid for the modified rules and tweaked tree structure) we may assume that any derivation of $\Gamma \vdash \Theta :- \mathtt{t}$ is free of (Cut_λ). The proof that such a derivation is uniform is by induction on its height.

Suppose $\mathtt{t}$ is not atomic and $\Gamma \vdash \Theta :- \mathtt{t}$ is the conclusion of any rule $(?L^-)$ other than (λR) or $(\cdot R)$. We may assume, by the induction hypothesis, that the derivations of the premises of $(?L^-)$ are uniform. The result follows by the induction hypothesis and the fact that (λR) or $(\cdot R)$ may be permuted with $(?L^-)$.

There are six permutations to consider, the three featuring (λR) are given in Figure 7. The remaining three cases featuring $(\cdot R)$ are straightforward. □

Remark 6.5 It is worth noting how the modified left rules are required for Theorem 6.4. In the cases of $(\rightsquigarrow L^-)$ and (λL^-), the restrictions ensure that they may be permuted with (λR) as shown in Figure 7. For example, looking at the third permutation of Figure 7 we see that after (λR) is swapped with $(\rightsquigarrow L^-)$, we must be sure that $\mathtt{x}$ is not free in Θ (otherwise the instance of (λR) is illegitimate). In the case of $(\cdot L^-)$ the restriction is so that it can always be permuted with $(\cdot R)$.

$$\dfrac{\dfrac{\begin{matrix}\Pi\\ \Gamma\vdash\langle\langle\ldots\langle\mathtt{t}_1,\mathtt{t}_2\rangle\ldots\rangle,\mathtt{x}\rangle\mathrel{:\!-}\mathtt{t}\end{matrix}}{\Gamma\vdash\langle\ldots\langle\mathtt{t}_1,\mathtt{t}_2\rangle\ldots\rangle\mathrel{:\!-}\lambda\mathtt{x}.\mathtt{t}}\,(\lambda R)}{\Gamma\vdash\langle\ldots\mathtt{t}_1{\cdot}\mathtt{t}_2\ldots\rangle\mathrel{:\!-}\lambda\mathtt{x}.\mathtt{t}}\,(\cdot L)\quad\longmapsto\quad\dfrac{\dfrac{\begin{matrix}\Pi\\ \Gamma\vdash\langle\langle\ldots\langle\mathtt{t}_1,\mathtt{t}_2\rangle\ldots\rangle,\mathtt{x}\rangle\mathrel{:\!-}\mathtt{t}\end{matrix}}{\Gamma\vdash\langle\ldots\mathtt{t}_1{\cdot}\mathtt{t}_2\ldots,\mathtt{x}\rangle\mathrel{:\!-}\mathtt{t}}\,(\cdot L)}{\Gamma\vdash\langle\ldots\mathtt{t}_1{\cdot}\mathtt{t}_2\ldots\rangle\mathrel{:\!-}\lambda\mathtt{x}.\mathtt{t}}\,(\lambda R)$$

$$\dfrac{\begin{matrix}\Pi_1\\ \Gamma\vdash\Theta\mathrel{:\!-}\mathtt{t}_1\end{matrix}\qquad\dfrac{\begin{matrix}\Pi_2\\ \Gamma\vdash\langle\langle\ldots\mathtt{t}_2[\mathtt{x}/\mathtt{t}_1]\ldots\rangle,\mathtt{x}\rangle\mathrel{:\!-}\mathtt{t}\end{matrix}}{\Gamma\vdash\langle\ldots\mathtt{t}_2[\mathtt{x}/\mathtt{t}_1]\ldots\rangle\mathrel{:\!-}\lambda\mathtt{x}.\mathtt{t}}\,(\lambda R)}{\Gamma\vdash\langle\ldots\langle\lambda\mathtt{x}.\mathtt{t}_2;\Theta\rangle\ldots\rangle\mathrel{:\!-}\lambda\mathtt{x}.\mathtt{t}}\,(\lambda L^-)$$

$$\longmapsto\quad\dfrac{\dfrac{\begin{matrix}\Pi_1\\ \Gamma\vdash\Theta\mathrel{:\!-}\mathtt{t}_1\end{matrix}\qquad\begin{matrix}\Pi_2\\ \Gamma\vdash\langle\langle\ldots\mathtt{t}_2[\mathtt{x}/\mathtt{t}_1]\ldots\rangle,\mathtt{x}\rangle\mathrel{:\!-}\mathtt{t}\end{matrix}}{\Gamma\vdash\langle\langle\ldots\langle\lambda\mathtt{x}.\mathtt{t}_2;\Theta\rangle\ldots\rangle,\mathtt{x}\rangle\mathrel{:\!-}\mathtt{t}}\,(\lambda L^-)}{\Gamma\vdash\langle\ldots\mathtt{t}_2[\mathtt{x}/\mathtt{t}_1]\ldots\rangle\mathrel{:\!-}\lambda\mathtt{x}.\mathtt{t}}\,(\lambda R)$$

$$\dfrac{\begin{matrix}\Pi_1\\ \Gamma\vdash\Theta\mathrel{:\!-}\mathtt{t}_1\end{matrix}\qquad\dfrac{\begin{matrix}\Pi_2\\ \Gamma\vdash\langle\langle\ldots\mathtt{t}_2\ldots\rangle,\mathtt{x}\rangle\mathrel{:\!-}\mathtt{t}\end{matrix}}{\Gamma\vdash\langle\ldots\mathtt{t}_2\ldots\rangle\mathrel{:\!-}\lambda\mathtt{x}.\mathtt{t}}\,(\lambda R)}{\Gamma,\mathtt{t}_1{\leadsto}\mathtt{t}_2\vdash\langle\ldots\Theta\ldots\rangle\mathrel{:\!-}\lambda\mathtt{x}.\mathtt{t}}\,({\leadsto}L^-)\quad\longmapsto\quad\dfrac{\dfrac{\begin{matrix}\Pi_1\\ \Gamma\vdash\Theta\mathrel{:\!-}\mathtt{t}_1\end{matrix}\qquad\begin{matrix}\Pi_2\\ \Gamma\vdash\langle\langle\ldots\mathtt{t}_2\ldots\rangle,\mathtt{x}\rangle\mathrel{:\!-}\mathtt{t}\end{matrix}}{\Gamma,\mathtt{t}_1{\leadsto}\mathtt{t}_2\vdash\langle\langle\ldots\Theta\ldots\rangle,\mathtt{x}\rangle\mathrel{:\!-}\mathtt{t}}\,({\leadsto}L^-)}{\Gamma\vdash\langle\ldots\mathtt{t}_2\ldots\rangle\mathrel{:\!-}\lambda\mathtt{x}.\mathtt{t}}\,(\lambda R)$$

$$\dfrac{\dfrac{\begin{matrix}\Pi_1\\ \Gamma\vdash\langle\ldots\langle\mathtt{t}_1;\mathtt{t}_2\rangle\ldots\rangle\mathrel{:\!-}\mathtt{s}_1\end{matrix}\qquad\begin{matrix}\Pi_2\\ \Gamma\vdash\Theta\mathrel{:\!-}\mathtt{s}_2\end{matrix}}{\Gamma\vdash\langle\langle\ldots\langle\mathtt{t}_1;\mathtt{t}_2\rangle\ldots\rangle,\Theta\rangle\mathrel{:\!-}\mathtt{s}_1{\cdot}\mathtt{s}_2}\,(\cdot R)}{\Gamma\vdash\langle\langle\ldots\langle\mathtt{t}_1{\cdot}\mathtt{t}_2\rangle\ldots\rangle,\Theta\rangle\mathrel{:\!-}\mathtt{s}_1{\cdot}\mathtt{s}_2}\,(\cdot L)\quad\longmapsto\quad\dfrac{\dfrac{\begin{matrix}\Pi_1\\ \Gamma\vdash\langle\ldots\langle\mathtt{t}_1;\mathtt{t}_2\rangle\ldots\rangle\mathrel{:\!-}\mathtt{s}_1\end{matrix}}{\Gamma\vdash\langle\ldots\langle\mathtt{t}_1{\cdot}\mathtt{t}_2\rangle\ldots\rangle\mathrel{:\!-}\mathtt{s}_1}\,(\cdot L)\qquad\begin{matrix}\Pi_2\\ \Gamma\vdash\Theta\mathrel{:\!-}\mathtt{s}_2\end{matrix}}{\Gamma\vdash\langle\langle\ldots\langle\mathtt{t}_1{\cdot}\mathtt{t}_2\rangle\ldots\rangle,\Theta\rangle\mathrel{:\!-}\mathtt{s}_1{\cdot}\mathtt{s}_2}\,(\cdot R)$$

Fig. 7. Permutations of left and right rules

To see the importance of the modified tree structure (the semicolons) note that $(\cdot R)$ generates new tree structure on the left of the term sequent. So if any subsequent instance of $(\cdot L)$ depends on this structure then it cannot be pushed behind $(\cdot R)$. The semicolons keep track of the tree structures on the left that depend only on (λL^-), the restriction on $(\cdot L^-)$ thus ensures that no instance of $(\cdot L^-)$ depends on a prior instance of $(\cdot R)$.

The restrictions on $({\leadsto}L^-)$, (λL^-) and $(\cdot L^-)$ and the modifications to the tree structure are sufficient but not necessary for Theorem 6.4. Refinements are for future research.

Restrictions arise in the notion of uniform derivation because — if we imagine a logic programming system based on these ideas — we must prevent the user from running programs which contain certain 'silly' reductions. This is not visible in the proof of cut-elimination, which considers the logic as a whole.

Definition 6.1 extends the existing notion of uniform derivation [8] to term sequent derivations. A term sequent derivation is uniform if (reading bottom up) we can always decompose the right part of a term sequent before decomposing the left parts: i.e. we can decompose the $\mathtt{t}$ of $\Gamma \vdash \Theta \mathrel{:\!-} \mathtt{t}$ before we decompose Θ. Compare with the more familiar notion of uniform derivation for sentence sequents 'a sentence sequent derivation is uniform when we can decompose the Δ of $\Gamma \vdash \Delta$ before we decompose Γ'.

Term sequent logic combines term sequents with sentence sequents, thus we consider a derivation uniform when its sentence sequent parts are uniform in the familiar sense, and its term sequent parts are uniform in the sense of Definition 6.1:

$$\frac{\Gamma \vdash \mathtt{t}_1 :- \mathtt{t}_2}{\Gamma \vdash \mathtt{t}_i :: \mathtt{t}_j}\,(E)\quad {\scriptstyle i=1, j=2 \text{ or } i=2, j=1}$$

$$\frac{\Gamma \vdash \mathtt{s}_1 :: \mathtt{t}_1 \quad \Gamma \vdash \mathtt{t}_2 :: \mathtt{s}_2}{\Gamma, \mathtt{t}_1 \approx \mathtt{t}_2 \vdash \mathtt{s}_1 :: \mathtt{s}_2}\,(\approx L) \qquad \frac{\Gamma \vdash \mathtt{t}_1 :: \mathtt{t}_2 \quad \ldots \quad \Gamma \vdash \mathtt{t}_{n-1} :: \mathtt{t}_n}{\Gamma \vdash \mathtt{t}_1 \approx \mathtt{t}_n, \Delta}\,(\approx R)$$

Fig. 8. Rules for λ-equality

Definition 6.6 A cut-free derivation of a sentence sequent is ***uniform*** if it is

(i) for every sentence sequent $\Gamma \vdash A$ in it, if A is not atomic the sequent is an instance of the conclusion of rule applying to A (a right rule).

(ii) every subderivation of a term sequent is uniform in the sense of Definition 6.1.

Theorem 6.7 *If $\Gamma \vdash \mathtt{A}$ is derivable in the single conclusion fragment of the term sequent calculus (i.e. where Δ is empty in any sentence sequent), then it is derivable by a uniform derivation.*

Proof. That the first condition on a uniform derivation is met is a well known result on intuitionistic logic, e.g. see [8]. That the second condition on a uniform derivation is met is proved by Theorem 6.4. □

7 λ-Equality

The equality offered by $\leftrightsquigarrow$ is strong (it relates relatively few terms). We often want two λ-terms equal when there is a chain of reductions linking them. To capture this we extend of term sequents with a new judgement form:

Definition 7.1 An ***equality-term sequent*** is a tuple $\Gamma \vdash \mathtt{t}_1 :: \mathtt{t}_2$ where $\mathtt{t}_1$ and $\mathtt{t}_2$ are terms.

We extend the syntax to include a binary atomic predicate $\approx$ and extend the term sequent rules by the rules of Figure 8.

Theorem 7.2 *If $\mathtt{x}$ does not occur in either argument position of $\rightsquigarrow$ in (any part of) the sentence $\mathtt{A}$ then $\Gamma \vdash \mathtt{A}[\mathtt{x}/\mathtt{t}_1], \Delta$ implies $\Gamma, \mathtt{t}_1 \approx \mathtt{t}_2 \vdash \mathtt{A}[\mathtt{x}/\mathtt{t}_2], \Delta$.*

Theorem 7.3 *$\vdash \mathtt{t}_1 :: \mathtt{t}_2$ if and only if $\mathtt{t}_1 = \mathtt{t}_2$, where $=$ is the reflexive-transitive-symmetric closure of $\twoheadrightarrow_{\alpha\beta\eta}$.*

Theorem 7.4 *Theorem 6.7 extends to a term sequent calculus extended to include $\approx$ and equality-term sequents.*

8 Conclusions

Related work.

Beeson's λ-logic [2] has the sentences of first-order logic with equality, and as term-language the λ-calculus. Our paper is in this spirit. In λ-logic there is no proof-theory for the term language. This paper gives a proof-theory for the λ-calculus with the advantage that consistency follows by purely proof-theoretic means.

Rewriting combines computation with logic: computation is expressed by rewrites; the logical judgement form is simply 's rewrites to t'. A literature of rewriting exists to prove confluence [1] which is comparable in richness and variety to that on cut-elimination. Also properties comparable to those following cut-elimination can be deduced from confluence. There are echoes of rewriting in our term logic for the λ-calculus — $\rightsquigarrow$ asserts a rewrite — yet the judgement form 's rewrites to t' is weaker than that of a typical logic. Notably, rewrites cannot be made conditional. Thus, cut-elimination for our system is a different and stronger result than a confluence proof. *Conditional* rewriting strengthens the judgement form [9], but 'if y reduces to $\lambda x.x$ then $y{\cdot}z$ reduces to z' still cannot be expressed or derived; see Derivation $D7$.

Deduction Modulo combines logical derivation with rewriting [6]. However, rewrites are imposed globally (we fix the rewrites, then do deduction 'modulo' those rewrites). Rewrites cannot be hypothesised by a sequent or made conditional. In term sequent logic, term sequents may be made conditional on complex predicates in the logic, as illustrated by our example derivations.

Deduction modulo is concerned with this issue and has a vocabulary to express the distinction: deduction modulo rewrite rules are *computations*, and term sequent equality is *deduction*. Enriching deduction modulo with deduction-style rewrites has been investigated [3]. This is known to be a non-trivial problem not yet fully resolved. Thus, this research is more general than deduction modulo and should be of interest to that community. However, term sequents apply to terms whereas deduction modulo rewrites can rewrite terms to terms, and *also* rewrite predicates to predicates. In that sense deduction modulo is more general than this research.

λ-prolog has λ-terms [11], which are typed. Terms are syntactically identified up to $\alpha\beta\eta$-equivalence and there is an equality up to $\alpha\beta\eta$ at every type. This is different from our system, which distinguishes sentences and terms, uses untyped terms, and does not syntactically identify terms up to $\alpha\beta\eta$. Still, it may be possible to make some connections.

The second author is known for studying variables and conditions on variables [5]. We note that the conditions on uniform proof (Definition 6.3) have to do with free variables. It is interesting to see conditions on free variables arise in this context.

Future work.

The idea of term sequents is novel. There seems no obstacle to taking the idea further. 'Ordinary' sentence sequents have been used in great variety to study different logics; a benefit of term sequents may be that they can also be used in great variety to study different kinds of equality on terms.

For example: Term sequent systems for integer and rational arithmetic will be studied in future publications. Furthermore, a much more general enquiry is possible to establish syntactic or semantic criteria on term sequent systems that guarantee cut-elimination.

The prospect of applying logical techniques to a variety of term systems is an interesting test of logical techniques: terms are *not* sentences; how far can we push logical techniques before they break? Ultimately, this may inform our understanding of 'what is a logic?' [7].

In this paper we have applied term sequents to build and study a model of functional programming within first order logic. First order logic itself has a well-known notion of computation given by uniform proof. The combination of the two in term sequent logic for the λ-calculus described in our notion of uniform proof, can be viewed as some kind of rewrite strategy — but currently we do not fully understand it. Matters are complicated because, as we have mentioned before in this paper, in term-sequents we can impose assumptions and thus, in effect, dynamically permit ‘extra rewrites’. It seems plausible that this could be the outline of a new and powerful programming environment. We detect shades of rewriting logic in this [10] but we cannot comment further on any connections. Careful further study of the system presented in this paper is justified.

References

[1] Franz Baader and Tobias Nipkow, *Term rewriting and all that*, Cambridge University Press, Great Britain, 1998.

[2] Michael Beeson, *Lambda logic*, 2nd Int'l Joint Conf. on Automated Reasoning (IJCAR 2004), LNCS, vol. 3097, Springer, 2004, pp. 460–474.

[3] Eric Deplagne and Claude Kirchner, *Deduction versus computation: the case of induction*, AISC and Calculemus Joint International Conferences, Lecture Notes in Artificial Intelligence, vol. 2385, Springer, July 2002, pp. 4–6.

[4] Murdoch J. Gabbay and Michael J. Gabbay, *a-logic with arrows*, Proceedings of WFLP, 2007, pp. 47–63.

[5] Murdoch J. Gabbay and A. M. Pitts, *A New Approach to Abstract Syntax with Variable Binding (journal version)*, Formal Aspects of Computing **13** (2001), no. 3–5, 341–363.

[6] Benjamin Werner Gilles Dowek, *Arithmetic as a theory modulo*, RTA, 2005, pp. 423–437.

[7] Ian Hacking, *What is logic?*, The Journal of Philosophy **76** (1979), 285–319.

[8] Joshua S. Hodas and Dale Miller, *Logic programming in a fragment of intuitionistic linear logic*, Proceedings 6th IEEE Annual Symp. on Logic in Computer Science, Amsterdam, The Netherlands, 15–18 July 1991, IEEE Computer Society Press, New York, 1991, pp. 32–42.

[9] Stephane Kaplan and Jean-Pierre Jouannaud (eds.), *Conditional term rewriting systems*, LNCS, vol. 308, Springer, July 1988.

[10] N. Marti-Oliet and J. Meseguer, *Rewriting logic as a logical and semantic framework*, ENTCS (J. Meseguer, ed.), vol. 4, Elsevier Science Publishers, 2000.

[11] Dale Miller, *A logic programming language with lambda-abstraction, function variables, and simple unification*, Extensions of Logic Programming **475** (1991), 253–281.

[12] Peter W. O'Hearn and David J. Pym, *The logic of bunched implications*, Bulletin of Symbolic Logic **2** (1999), no. 5, 215–244.