# Strongly Normalizing Higher-Order Relational Queries


## Wilmer Ricciotti

Laboratory for Foundations of Computer Science, University of Edinburgh, United Kingdom
http://www.wilmer-ricciotti.net
research@wilmer-ricciotti.net

## James Cheney

Laboratory for Foundations of Computer Science, University of Edinburgh, United Kingdom
https://homepages.inf.ed.ac.uk/jcheney/
jcheney@inf.ed.ac.uk



## Abstract

Language-integrated query is a powerful programming construct allowing database queries and ordinary program code to interoperate seamlessly and safely. Language-integrated query techniques rely on classical results about monadic comprehension calculi, including the *conservativity theorem* for nested relational calculus. Conservativity implies that query expressions can freely use nesting and unnesting, yet as long as the query result type is a flat relation, these capabilities do not lead to an increase in expressiveness over flat relational queries. Wong showed how such queries can be translated to SQL via a constructive rewriting algorithm, and Cooper and others advocated *higher-order* nested relational calculi as a basis for language-integrated queries in functional languages such as Links and F#. However there is no published proof of the central *strong normalization* property for higher-order nested relational queries: a previous proof attempt does not deal correctly with rewrite rules that duplicate subterms. This paper fills the gap in the literature, explaining the difficulty with a previous proof attempt, and showing how to extend the $\top\top$-*lifting* approach of Lindley and Stark to accommodate duplicating rewrites. We also sketch how to extend the proof to a recently-indroduced calculus for *heterogeneous* queries mixing set and multiset semantics.



**2012 ACM Subject Classification** Theory of computation → Type theory

**Keywords and phrases** Strong normalization, $\top\top$-lifting, Nested relational calculus, Language-integrated query

**Digital Object Identifier** 10.4230/LIPIcs...


## 1 Introduction

The nested relational calculus [2] provides a principled foundation for integrating database queries into programming languages. Wong's conservativity theorem [20] generalized the classic flat-flat theorem [17] to show that for any nesting depth $d$, a query expression over flat input tables returning collections of depth at most $d$ can be expressed without constructing intermediate results of nesting depth greater than $d$. In the special case $d = 1$, this implies the flat-flat theorem, namely that a nested relational query mapping flat tables to flat tables can be expressed equivalently using the flat relational calculus. In addition, Wong's proof technique was constructive, and gave an easily-implemented terminating rewriting algorithm for normalizing NRC queries to equivalent flat queries; these normal forms correspond closely to idiomatic SQL queries and translating from the former to the latter is straightforward. The basic approach has been extended in a number of directions, including to allow for (nonrecursive) higher-order functions in queries [7], and to allow for translating queries that return nested results to a bounded number of flat relational queries [4].

Normalization-based techniques are used in language-integrated query systems such as Kleisli [21] and Links [8], and can improve both performance and reliability of language-integrated query in F# [3]. However, most work on normalization considers *homogeneous* queries in which there is a single collection type (e.g. homogeneous sets or multisets). Recently, we considered a *heterogeneous* calculus for mixed set and bag queries [19], and conjectured that it too satisfies strong normalization and conservativity theorems. However, in attempting to extend Cooper's proof of normalization we discovered a subtle problem, which makes the original proof incomplete.

Most techniques to prove the strong normalization property for higher-order languages employ logical relations; among these, the Girard-Tait *reducibility* relation is particularly influential: reducibility interprets types as certain sets of strongly normalizing terms enjoying certain closure properties with respect to reduction, called *candidates of reducibility* [9]. The fundamental theorem then proves that every well-typed term is reducible, hence also strongly normalizing. In its traditional form, reducibility has a limitation that makes it difficult to apply it to certain calculi: the elimination form of every type is expected to be a *neutral* term or, informally, an expression that, when placed in an arbitrary evaluation context, does not interact with it by creating new redexes. However, certain calculi possess *commuting conversions*, i.e. reduction rules that apply to nested elimination forms: such rules usually arise when the elimination form for a type $S$ is constructed by means of a term of an arbitrary type $T$, unrelated to $S$. In this case, we expect nested elimination forms to commute; for example, given terms $s$ of type $S$ and $t$ of type $T$, an elimination context $\mathcal{E}_T$ for terms of type $T$, and an elimination context $\mathcal{E}_S$ for terms of type $S$ indexed by terms of an arbitrary type, we could have the following commuting conversion:

$$\mathcal{E}_T[\mathcal{E}_S(t)[s]] \rightsquigarrow \mathcal{E}_S(\mathcal{E}_T[t])[s]$$

Since in the presence of commuting conversions elimination forms are not neutral, a straightforward adaptation of reducibility to such languages is precluded.

## 1.1 $\top\top$-lifting and $NRC_\lambda$

Cooper's $NRC_\lambda$ [6, 7] extends the simply typed lambda calculus with collection types whose elimination form is expressed by *comprehensions* $\bigcup\{M|x \leftarrow N\}$, where $M$ and $N$ have a collection type, and the bound variable $x$ can appear in $M$:

$$\frac{\Gamma \vdash N : \{S\} \qquad \Gamma, x : S \vdash M : \{T\}}{\Gamma \vdash \bigcup\{M|x \leftarrow N\} : \{T\}}$$

This comprehension destructures collections of type $\{S\}$ to produce new collections in $\{T\}$, where $T$ is an unrelated type: semantically, this corresponds to the union of all the collections $M\,[V/x]$, such that $V$ is in $N$. According to the standard approach, we should attempt to define the reducibility predicate for $\{S\}$ as:

$$\mathsf{Red}_{\{S\}} \triangleq \{N : \forall x, T, \forall M \in \mathsf{Red}_{\{T\}}, \bigcup\{M|x \leftarrow N\} \in \mathsf{Red}_{\{T\}}\}$$

(we use a typewriter style $\{\cdot\}$ for collections as terms of $NRC_\lambda$, to distinguish them from metalinguistic sets $\{\cdot\}$). Of course the definition above is circular, since it uses reducibility over collections to express reducibility over collections; however, this inconvenience could in principle be circumvented by means of impredicativity, replacing $\mathsf{Red}_{\{T\}}$ with a suitable, universally quantified candidate of reducibility (an approach we used in [18] in the context of justification logic). Unfortunately, the arbitrary return type of comprehensions is not the only problem: they are also involved in commuting conversions, such as:

$$\bigcup\{M|x \leftarrow \bigcup\{N|y \leftarrow P\}\} \rightsquigarrow \bigcup\{\bigcup\{M|x \leftarrow N\}|y \leftarrow P\} \qquad (y \notin FV(M))$$

Because of this rule, comprehensions are not neutral terms, thus we cannot use the closure properties of candidates of reducibility (in particular, CR3) to prove that a collection term is reducible. To address this problem, Lindley and Stark proposed a revised notion of reducibility based on a technique they called $\top\top$-lifting [15], which involves quantification over arbitrarily nested, reducible elimination contexts (*continuations*). The technique is actually composed of two steps: $\top$-lifting, used to define the set $\mathsf{Red}_T^\top$ of reducible continuations of type $T$ in terms of $\mathsf{Red}_T$, and $\top\top$-lifting proper, defining $\mathsf{Red}_{\{T\}} = \mathsf{Red}_T^{\top\top}$ in terms of $\mathsf{Red}_T^\top$. In our setting, we would have:

$$\mathsf{Red}_T^\top \triangleq \{K : \forall M \in \mathsf{Red}_T, K[\{M\}] \in \mathcal{SN}\}$$
$$\mathsf{Red}_T^{\top\top} \triangleq \{M : \forall K \in \mathsf{Red}_T^\top, K[M] \in \mathcal{SN}\}$$

In $NRC_\lambda$, however, we come across an additional problem concerning the property of distributivity of comprehensions over unions, represented by the following reduction rule:

$$\bigcup\{M \cup N | x \leftarrow P\} \leadsto \bigcup\{M | x \leftarrow P\} \cup \bigcup\{N | x \leftarrow P\}$$

One can immediately see that in $\bigcup\{M \cup N | x \leftarrow \square\}$ the reduction above duplicates the hole, producing a multi-hole context that is not a continuation in the Lindley-Stark sense.

Cooper in his work attempted to reconcile continuations with duplicating reductions. While considering extensions to his language, we discovered that his proof of strong normalization presents a nontrivial lacuna which we could only fix by relaxing the definition of continuations to allow multiple holes. This problem affected both the proof of the original result and our attempt to extend it, and has an avalanche effect on definitions and proofs, yielding a more radical revision of the $\top\top$-lifting technique which is the subject of this paper.

The contribution of this paper is to place previous work on higher-order programming for language-integrated query on a solid foundation. As we will show, our approach also extends to prove normalization for a higher-order heterogeneous collection calculus $NRC_\lambda(Set, Bag)$ [19] and we believe our proof technique can be extended further.

## 1.2 Summary

Section 2 reviews presents $NRC_\lambda$ and its rewrite system. Section 3 presents the refined approach to reducibility needed to handle rewrite rules with branching continuations. Section 4 presents the proof of strong normalization for $NRC_\lambda$. Section 5 outline the extension to a higher-order calculus $NRC_\lambda(Set, Bag)$ providing heterogeneous set and bag queries. Sections 6 and 7 discuss related work and conclude. Some of the proofs which were omitted from the paper due to space constraints and are detailed in the appendix.

# 2 Higher-order NRC

$NRC_\lambda$, a nested relational calculus with non-recursive higher order functions, is defined by the following grammar:

$$\begin{array}{llcl}
\textbf{types} & S, T & ::= & A \mid S \to T \mid \langle \overrightarrow{\ell : T} \rangle \mid \{T\} \\
\textbf{terms} & L, M, N & ::= & x \mid c(\overrightarrow{M}) \mid \langle \overrightarrow{\ell = M} \rangle \mid M.\ell \mid \lambda x.M \mid (M\ N) \\
 & & \mid & \emptyset \mid \{M\} \mid M \cup N \mid \bigcup\{M | x \leftarrow N\} \\
 & & \mid & \texttt{empty}\ M \mid \texttt{where}\ M\ \texttt{do}\ N
\end{array}$$

Types include atomic types $A, B, \ldots$ (among which we have Booleans $\mathbf{B}$), record types with named fields $\langle \overrightarrow{\ell : T} \rangle$, collections $\{T\}$; we define *relation types* as those in the form

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T} \qquad \frac{\Sigma(c) = \overrightarrow{A_n} \to A' \qquad (\Gamma \vdash M_i : A_i)_{i=1,\dots,n}}{\Gamma \vdash c(\overrightarrow{M_n}) : A'}$$

$$\frac{(\Gamma \vdash M_i : T_i)_{i=1,\dots,n}}{\Gamma \vdash \langle \overrightarrow{\ell_n = M_n} \rangle : \langle \overrightarrow{\ell_n : T_n} \rangle} \qquad \frac{\Gamma \vdash M : \langle \overrightarrow{\ell_n : T_n} \rangle \qquad i \in \{1, \dots, n\}}{\Gamma \vdash M.\ell_i : T_i}$$

$$\frac{\Gamma, x : S \vdash M : T}{\Gamma \vdash \lambda x.M : S \to T} \qquad \frac{\Gamma \vdash M : S \to T \qquad \Gamma \vdash N : S}{\Gamma \vdash (M\ N) : T}$$

$$\frac{}{\Gamma \vdash \emptyset : \{T\}} \qquad \frac{\Gamma \vdash M : T}{\Gamma \vdash \{M\} : \{T\}} \qquad \frac{\Gamma \vdash M : \{T\} \qquad \Gamma \vdash N : \{T\}}{\Gamma \vdash M \cup N : \{T\}}$$

$$\frac{\Gamma, x : T \vdash M : \{S\} \qquad \Gamma \vdash N : \{T\}}{\Gamma \vdash \bigcup \{M | x \leftarrow N\} : \{S\}}$$

$$\frac{\Gamma \vdash M : \{T\}}{\Gamma \vdash \mathtt{empty}\ M : \mathbf{B}} \qquad \frac{\Gamma \vdash M : \mathbf{B} \qquad \Gamma \vdash N : \{T\}}{\Gamma \vdash \mathtt{where}\ M\ \mathtt{do}\ N : \{T\}}$$

**Figure 1** Type system of $NRC_\lambda$.

$\{\langle \overrightarrow{\ell : A} \rangle\}$, i.e. collections of tuples of atomic types. Terms include applied constants $c(\overrightarrow{M})$, records with named fields and record projections ($\langle \ell = M \rangle$, $M.\ell$), various collection terms (empty, singleton, union, and comprehension), the emptiness test `empty`, and one-sided conditional expressions for collection types `where` $M$ `do` $N$. In this definition, $x$ ranges over variable names, $c$ over constants, and $\ell$ over record field names. We will allow ourselves to use sequences of generators in comprehensions, which are syntactic sugar for nested comprehensions, e.g.:

$$\bigcup \{M | x \leftarrow N, y \leftarrow R\} \triangleq \bigcup \{\bigcup \{M | y \leftarrow R\} | x \leftarrow N\}$$

The typing rules, shown in Figure 1, are largely standard, and we only mention those operators that are specific to our language: constants are typed according to a fixed signature $\Sigma$, prescribing the types of the $n$ arguments and of the returned expression to be atomic; `empty` takes a collection and returns a Boolean indicating whether its argument is empty; `where` takes a Boolean condition and a collection and returns the second argument if the Boolean is true, otherwise the empty set. (Conventional two-way conditionals, at any type, are omitted for convenience but can be added without difficulty.)

## 2.1 Reduction and normalization

$NRC_\lambda$ is equipped with a rewrite system whose purpose is to convert expressions of flat relation type into a sublanguage isomorphic to a fragment of SQL, even when the original expression contains subterms whose type is not available in SQL, such as nested collections. The rules for this rewrite system are shown in Figure 2.

Reduction on applied constants can happen when all of the arguments are in normal form, and relies on a fixed semantics $[\![\cdot]\!]$ which assigns to each constant $c$ of signature $\Sigma(c) = \overrightarrow{A_n} \to A'$ a function mapping sequences of values of type $\overrightarrow{A_n}$ to values of type $A'$. The rules for collections and conditionals are mostly standard. The reduction rule for the emptiness test is triggered when the argument $M$ is not of relation type (but, for instance, of nested collection type) and employs comprehension to generate a (trivial) relation that is empty if and only if $M$ is.

$$(\lambda x.M)\ N \rightsquigarrow M[N/x] \qquad \langle \dots, \ell = M, \dots \rangle.\ell \rightsquigarrow M \qquad c(\overrightarrow{V}) \rightsquigarrow [\![c]\!]\,(\overrightarrow{V})$$

$$\begin{array}{rcll}
\bigcup\{\emptyset | x \leftarrow M\} \rightsquigarrow \emptyset \quad \bigcup\{M | x \leftarrow \emptyset\} \rightsquigarrow \emptyset & & \bigcup\{M | x \leftarrow \{N\}\} \rightsquigarrow M[N/x] & \\
\bigcup\{M \cup N | x \leftarrow R\} & \rightsquigarrow & \bigcup\{M | x \leftarrow R\} \cup \bigcup\{N | x \leftarrow R\} & \\
\bigcup\{M | x \leftarrow N \cup R\} & \rightsquigarrow & \bigcup\{M | x \leftarrow N\} \cup \bigcup\{M | x \leftarrow R\} & \\
\bigcup\{M | y \leftarrow \bigcup\{R | x \leftarrow N\}\} & \rightsquigarrow & \bigcup\{M | x \leftarrow N, y \leftarrow R\} & (\text{if } x \notin \mathrm{FV}(M)) \\
\bigcup\{M | x \leftarrow \mathtt{where}\ N\ \mathtt{do}\ R\} & \rightsquigarrow & \bigcup\{\mathtt{where}\ N\ \mathtt{do}\ M | x \leftarrow R\} & (\text{if } x \notin \mathrm{FV}(M))
\end{array}$$

$$\begin{array}{rcll}
\mathtt{where}\ \mathsf{true}\ \mathtt{do}\ M \rightsquigarrow M \quad \mathtt{where}\ \mathsf{false}\ \mathtt{do}\ M \rightsquigarrow \emptyset & & \mathtt{where}\ M\ \mathtt{do}\ \emptyset \rightsquigarrow \emptyset & \\
\mathtt{where}\ M\ \mathtt{do}\ (N \cup R) & \rightsquigarrow & (\mathtt{where}\ M\ \mathtt{do}\ N) \cup (\mathtt{where}\ M\ \mathtt{do}\ R) & \\
\mathtt{where}\ M\ \mathtt{do}\ \bigcup\{N | x \leftarrow R\} & \rightsquigarrow & \bigcup\{\mathtt{where}\ M\ \mathtt{do}\ N | x \leftarrow R\} & \\
\mathtt{where}\ M\ \mathtt{do}\ \mathtt{where}\ N\ \mathtt{do}\ R & \rightsquigarrow & \mathtt{where}\ (M \wedge N)\ \mathtt{do}\ R & \\
\mathtt{empty}\ M & \rightsquigarrow & \mathtt{empty}\ (\bigcup\{\langle\rangle | x \leftarrow M\}) & (\text{if } M \text{ is not relation-typed})
\end{array}$$

**Figure 2** Query normalization

The normal forms of queries under these rewriting rules construct no intermediate nested structures, and are straightforward to translate to equivalent SQL queries. Cooper [7] and Lindley and Cheney [14] give details of such translations. Cheney et al. [3] showed how to improve the performance and reliability of LINQ in F# using normalisation and gave many examples showing how higher-order queries support a convenient, compositional language-integrated query programming style.

## 3 Reducibility with branching continuations

We introduce here the extension of ⊤⊤-lifting we use to derive a proof of strong normalization for $NRC_\lambda$. The main contribution of this section is a refined definition of continuations with branching structure and multiple holes, as opposed to the linear structure with a single hole used by standard ⊤⊤-lifting. In our definition, continuations (as well as the more general notion of context) are particular forms of terms: in this way, the notion of term reduction can be used for continuations as well, without need for auxiliary definitions.

### 3.1 Contexts and continuations

We start our discussion by introducing *contexts*, or terms with multiple, labelled holes that can be instantiated by plugging other terms (including other contexts) into them.

▶ **Definition 1** (context)**.** *Let us fix a countably infinite set $\mathcal{P}$ of indices: a* context *$C$ is a term that may contain distinguished free variables $[p]$, also called* holes, *where $p \in \mathcal{P}$.*

*Given a finite map from indices to terms $[p_1 \mapsto M_1, \dots, p_n \mapsto M_n]$ (*context instantiation*), the notation $C[p_1 \mapsto M_1, \dots, p_n \mapsto M_n]$ (*context application*) denotes the term obtained by simultaneously substituting $M_1, \dots, M_n$ for the holes $[p_1], \dots, [p_n]$.*

*We will use metavariables $\eta, \theta$ to denote context instantiations.*

▶ **Definition 2** (support)**.** *Given a context $C$, its* support $\mathsf{supp}(C)$ *is defined as the set of the indices $p$ such that $[p]$ occurs in $C$ as a free variable:*

$$\mathsf{supp}(C) \triangleq \{p : [p] \in \mathrm{FV}(C)\}$$

When a term does not contain any $[p]$, we say that it is a *pure* term; when it is important that a term be pure, we will refer to it by using overlined metavariables $\overline{L}, \overline{M}, \overline{N}, \overline{R}, \dots$. Under the appropriate assumptions, a multiple context instantiation can be decomposed.

▶ **Definition 3.** *An instantiation* $\eta$ *is* permutable *iff for all* $p \in \mathrm{dom}(\eta)$ *we have* $\mathrm{FV}(\eta(p)) \cap \mathrm{dom}(\eta) = \emptyset$.

▶ **Lemma 4.** *Let* $\eta$ *be permutable and let us denote by* $\eta_{\neg p}$ *the restriction of* $\eta$ *to indices other than* $p$*. Then for all* $p \in \mathrm{dom}(\eta)$ *we have:*

$$C\eta = C[p \mapsto \eta(p)]\eta_{\neg p} = C\eta_{\neg p}[p \mapsto \eta(p)]$$

We can now define continuations as certain contexts that capture how one or more collections can be used in a program.

▶ **Definition 5** (continuation)**.** *Continuations* $K$ *are defined as the following subset of contexts:*

$$K, H ::= \quad [p] \mid \overline{M} \mid K \cup K \mid \bigcup\{\overline{M}|x \leftarrow K\} \mid \texttt{where}\ \overline{B}\ \texttt{do}\ K$$

*where for all indices* $p$*,* $[p]$ *can occur at most once.*

This definition differs from the traditional one in two ways: first, holes are decorated with an index; secondly, and most importantly, the production $K \cup K$ allows continuations to branch and, as a consequence, to use more than one hole. Note that the grammar above is ambiguous, in the sense that certain expressions like $\texttt{where}\ \overline{B}\ \texttt{do}\ \overline{N}$ can be obtained either from the production $\texttt{where}\ \overline{B}\ \texttt{do}\ K$ with $K = N$, or as pure terms by means of the production $\overline{M}$: we resolve this ambiguity by parsing these expressions as pure terms whenever possible, and as continuations only when they are proper continuations. An additional complication of $NRC_\lambda$ when compared to the computational metalanguage for which $\top\top$-lifting was devised lies in the way conditional expressions can reduce when placed in an arbitrary context: continuations in the grammar above are not liberal enough to adapt to such reductions, therefore, like Cooper, we will need an additional definition of *auxiliary* continuations allowing holes to appear in the body of a comprehension (in addition to comprehension generators).

▶ **Definition 6** (auxiliary continuation)**.** *Auxiliary continuations are defined as the following subset of contexts:*

$$Q, O ::= \quad [p] \mid \overline{M} \mid Q \cup Q \mid \bigcup\{Q|x \leftarrow Q\} \mid \texttt{where}\ \overline{B}\ \texttt{do}\ Q$$

*where for all indices* $p$*,* $[p]$ *can occur at most once.*

A continuation is therefore a special case of auxiliary continuation; however, an auxiliary continuation is allowed to branch not only with unions, but also with comprehensions.[1] We use the following definition of *frames* to represent flat continuations with a distinguished hole.

▶ **Definition 7** (frame)**.** *Frames are defined by the following grammar:*

$$F ::= \quad \bigcup\{Q|x\} \mid \bigcup\{x \leftarrow Q\} \mid \texttt{where}\ \overline{B}$$

*where for all indices* $p$*,* $[p]$ *can occur at most once.*

[1] It is worth noting that Cooper's original definition of auxiliary continuation does not use branching comprehension (nor branching unions), but is linear just like the original definition of continuation. The only difference between regular and auxiliary continuations in his work is that the latter allowed nesting not just within comprehension generators, but also within comprehension bodies (in our notation, this would correspond to two separate productions $\bigcup\{\overline{M}|x \leftarrow Q\}$ and $\bigcup\{Q|x \leftarrow \overline{N}\}$).

*The operation $F^p$, lifting a frame to an auxiliary continuation with a distinguished hole $[p]$ is defined by the following rules*

$$\begin{array}{lll} \bigcup\{Q|x\}^p & = \bigcup\{Q|x \leftarrow [p]\} & (p \notin \mathsf{supp}(Q)) \\ \bigcup\{x \leftarrow Q\}^p & = \bigcup\{[p]\,|x \leftarrow Q\} & (p \notin \mathsf{supp}(Q)) \\ (\texttt{where}\ B)^p & = \texttt{where}\ B\ \texttt{do}\ [p] & \end{array}$$

*The composition operation $Q \circledcirc_p F$ is defined as:*

$$Q \circledcirc_p F = Q[p \mapsto F^p]$$

We generally use frames in conjunction with continuations or auxiliary continuations when we need to partially expose their leaves: for example, if we write $K = K_0 \circledcirc_p \bigcup\{\overline{M}|x\}$, we know that instantiating $K$ at index $p$ with (for example) a singleton will create a redex: $K[p \mapsto \{\overline{L}\}] \rightsquigarrow K_0[p \mapsto \overline{M}\,[\overline{L}/x]]$.

We introduce two measures $|\cdot|_p$ and $\|\cdot\|_p$ denoting the nesting depth of a hole $[p]$: the two measures differ in the treatment of nesting within the body of a comprehension.

▶ **Definition 8.** *The measures $|Q|_p$ and $\|Q\|_p$ are defined as follows:*

$$|[q]|_p = \|[q]\|_p = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{else} \end{cases}$$

$$|\overline{M}|_p = \|\overline{M}\|_p = 0$$

$$|Q_1 \cup Q_2|_p = \max(|Q_1|_p, |Q_2|_p) \qquad \|Q_1 \cup Q_2\|_p = \max(\|Q_1\|_p, |Q_2|_p)$$

$$|\texttt{where}\ B\ Q|_p = |Q|_p + 1 \qquad \|\texttt{where}\ B\ Q\|_p = \|Q\|_p + 1$$

$$|\bigcup\{Q_1|x \mapsto Q_2\}|_p = \begin{cases} |Q_1|_p & \text{if } p \in \mathsf{supp}(Q_1) \\ |Q_2|_p + 1 & \text{if } p \in \mathsf{supp}(Q_2) \\ 0 & \text{else} \end{cases}$$

$$\|\bigcup\{Q_1|x \mapsto Q_2\}\|_p = \begin{cases} \|Q_1\|_p + 1 & \text{if } p \in \mathsf{supp}(Q_1) \\ \|Q_2\|_p + 1 & \text{if } p \in \mathsf{supp}(Q_2) \\ 0 & \text{else} \end{cases}$$

$NRC_\lambda$ reduction can be used immediately on contexts (including regular and auxiliary continuations) since these are simply terms with distinguished free variables; we will also abuse notation to allow ourselves specify reduction on hole instantiations: whenever $\eta(p) \rightsquigarrow N$ and $\eta' = \eta_{\neg p}[p \mapsto N]$, we can write $\eta \rightsquigarrow \eta'$.

We will denote the set of strongly normalizing terms by $\mathcal{SN}$. For strongly normalizing terms (and by extension for hole instantiations containing only strongly normalizing terms), we can introduce the concept of maximal reduction length.

▶ **Definition 9** (maximal reduction length)**.** *Suppose $M \in \mathcal{SN}$: then , we define $\nu(M)$ as the maximum length of all reduction sequences starting with $M$.*

▶ **Lemma 10.** *For all strongly normalizing terms $M$, if $M \rightsquigarrow M'$, then $\nu(M') < \nu(M)$.*

With an abuse of notation, given a context application $\eta$, we write $\nu(\eta)$ for $\sum_{p \in \mathrm{dom}(\eta)} \nu(\eta(p))$ (whenever this value is defined).

## 3.2 Renaming reduction

Reducing a plain or auxiliary continuation will yield a context that is not necessarily in the same class because certain holes may have been duplicated. For this reason, we introduce a refined notion of renaming reduction which we can use to rename holes in the results so that each of them occurs at most one time.

▶ **Definition 11.** *Given a term with holes $M$ and a finite map $\sigma : \mathcal{P} \to \mathcal{P}$, we write $M\sigma$ for the term obtained from $M$, replacing each hole $[p]$ such that $\sigma(p)$ is defined with $[\sigma(p)]$.*

Even though finite renaming maps are partial functions, it is convenient to extend them to total functions by taking $\sigma(p) = p$ whenever $p \notin \mathrm{dom}(\sigma)$; we will write $\mathsf{id}$ to denote the empty renaming map, whose total extension is the identity function on $\mathcal{P}$.

▶ **Definition 12** (renaming reduction)**.** *$M$ $\sigma$-reduces to $N$ (notation: $M \overset{\sigma}{\rightsquigarrow} N$) iff $M \rightsquigarrow N\sigma$.*

Conveniently, it can be shown that every renaming reduction chain can be simulated by a plain reduction chain of the same length and vice-versa: therefore the notion of strongly normalizing term and the maximal reduction length $\nu(M)$ do not depend on whether we use plain or renaming reduction (this simple result is described in the appendix).

Our goal is to describe the reduction of pure terms expressed in the form of instantiated continuations. One first difficulty we need to overcome is that, as we noted, the sets of continuations (both regular and auxiliary) are not closed under reduction; thankfully, we can prove they are closed under renaming reduction.

▶ **Lemma 13.**

1. *For all continuations $K$, if $K \rightsquigarrow C$, there exist a continuation $K'$ and a finite map $\sigma$ such that $K \overset{\sigma}{\rightsquigarrow} K'$ and $K'\sigma = C$.*
2. *For all auxiliary continuations $Q$, if $Q \rightsquigarrow C$, there exist an auxiliary continuation $Q'$ and a finite map $\sigma$ such that $Q \overset{\sigma}{\rightsquigarrow} Q'$ and $Q'\sigma = C$.*

**Proof sketch.** For all $C$ we can find $C', \sigma$ such that $C = C'\sigma$ and all the holes in $C'$ are linear. For case 1, we can show by induction on the derivation of $K \rightsquigarrow C'\sigma$ that $C'$ satisfies the grammar for continuations. Case 2 is similar. ◀

Secondly, given a renaming reduction $C \overset{\sigma}{\rightsquigarrow} C'$, we want to be able to express the corresponding reduction on $C\eta$: due to the renaming $\sigma$, it is not enough to change $C$ to $C'$, but we also need to construct some $\eta'$ containing precisely those mappings $[q \mapsto M]$ such that, if $\sigma(q) = p$, then $p \in \mathrm{dom}(\eta)$ and $\eta(p) = M$. This construction is expressed by means of the following operation.

▶ **Definition 14.** *For all pure hole instantiations $\eta$ and renamings $\sigma$, we define $\eta^\sigma$ as the hole instantiation such that:*

- *if $\sigma(p) \in \mathrm{dom}(\eta)$ then $\eta^\sigma(p) = \eta(\sigma(p))$;*
- *in all other cases, $\eta^\sigma(p) = \eta(\sigma)$.*

The results above allow us to express what happens when a reduction duplicates the holes in a continuation which is then combined with a hole instantiation.

▶ **Lemma 15.** *For all contexts $C$, renamings $\sigma$, and hole instantiations $\eta$ such that, for all $p \in \mathrm{dom}(\eta)$, $\mathsf{supp}(\eta(p)) \cap \mathrm{dom}(\sigma) = \emptyset$, if $C \overset{\sigma}{\rightsquigarrow} C'$, then $C\eta \overset{\sigma}{\rightsquigarrow} C'\eta^\sigma$.*

**Remark.** In [6], Cooper attempts to prove strong normalization for $NRC_\lambda$ using a similar, but weaker result:

> If $K \rightsquigarrow C$, then for all terms $M$ there exists $K'_M$ such that $C[M] = K'_M[M]$ and $K[M] \rightsquigarrow K'_M[M]$.

Since he does not have multi-hole continuations and renaming reductions, his reasoning is that, whenever a hole is duplicated, e.g.

$$K = \bigcup \{N_1 \cup N_2 | x \leftarrow \Box\} \rightsquigarrow \bigcup \{N_1 | x \leftarrow \Box\} \cup \bigcup \{N_2 | x \leftarrow \Box\} = C$$

he resorts to obtaining a continuation from $C$ simply by filling one of the holes with the instantiation $M$:

$$K'_M = \bigcup \{N_1 | x \leftarrow M\} \cup \bigcup \{N_2 | x \leftarrow \Box\}$$

Hence, $K'_M[M] = C[M]$. Unfortunately, subsequent proofs rely on the fact that $\nu(K)$ must decrease under reduction: since we have no control over $\nu(M)$, which could potentially be much greater than $\nu(K)$, it may be that $\nu(K'_M) \geq \nu(K)$.

In our setting, by combining Lemma 13 and 15, we can find a $K'$ which is a proper contractum of $K$, so that $\nu(K') < \nu(K)$, as required by subsequent proofs.

The following result, like many other in the rest of this section, proceeds by well-founded induction; we will use the following notation to represent well-founded relations:

- $<$ stands for the standard less-than relation on $\mathbb{N}$, which is well-founded;
- $\lessdot$ is the lexicographic extension of $<$ to $\mathbb{N}^k$, also well-founded;
- $\prec$ will be used to provide a decreasing metric that depends on the specific proof: such metrics are defined as subsets of $\lessdot$ and are thus well-founded.

▶ **Lemma 16.** *Let $Q$ be an auxiliary continuation, and let $\eta, \theta$ context instantiations s.t. their union is permutable. If $Q\eta \in \mathcal{SN}$ and $Q\theta \in \mathcal{SN}$, then $Q\eta\theta \in \mathcal{SN}$.*

**Proof.** We assume, that all of the instantiations in $\eta$ and $\theta$ are effective (otherwise, we can find strictly smaller $\eta', \theta'$ such that $Q\eta\theta = Q\eta'\theta'$, and all the instantiations are effective). We show $Q\eta \in \mathcal{SN}$ and $Q\theta \in \mathcal{SN}$ imply $Q \in \mathcal{SN}$, $\eta \in \mathcal{SN}$ and $\theta \in \mathcal{SN}$; thus we can then prove the theorem by well-founded induction on $(Q, \eta, \theta)$ using the following metric:

$$(Q_1, \eta_1, \theta_1) \prec (Q_2, \eta_2, \theta_2) \iff (\nu(Q_1), \|Q_1\|, \nu(\eta_1) + \nu(\theta_1)) \lessdot (\nu(Q_2), \|Q_2\|, \nu(\eta_2) + \nu(\theta_2))$$

We show that all of the possible contracta of $Q\eta\theta$ are s.n. by case analysis on the contraction. The important cases are the following:

- $Q'\eta^\sigma\theta^\sigma$, where $Q \overset{\sigma}{\rightsquigarrow} Q'$: it is easy to see that $\nu(\eta^\sigma)$ and $\nu(\theta^\sigma)$ are defined because $\nu(\eta)$ and $\nu(\theta)$ are; then the thesis follows from induction hypothesis, knowing that $\nu(Q') < \nu(Q)$ (Lemma 10).
- $Q_0[p \mapsto N]\eta_0\theta$ where $Q = Q_0 \textcircled{p} F$, $\eta = [p \mapsto M]\eta_0$, and $F^p[p \mapsto M] \rightsquigarrow N$ (reduction at the interface). By Lemma 51 we know $\nu(Q_0) \leq \nu(Q)$; we can easily prove $\|Q_0\| < \|Q\|$; we take $\eta' = [p \mapsto N]\eta_0$: since $Q\eta$ reduces to $Q_0\eta'$ and both terms are strongly normalizing, we have that $\nu(\eta')$ is defined. Then observe $(Q_0, \eta', \theta) \prec (Q, \eta, \theta)$ and obtain the thesis by induction hypothesis. A symmetric case with $p \in \mathrm{dom}(\theta)$ is proved similarly. ◀

▶ **Corollary 17.** *$Q[p \mapsto M]^\sigma \in \mathcal{SN}$ iff for all $q$ s.t. $\sigma(q) = p$, we have $Q[q \mapsto M] \in \mathcal{SN}$.*

## 3.3 Candidates of reducibility

We here define the notion of *candidates of reducibility*: sets of strongly normalizing terms enjoying certain closure properties that can be used to overapproximate the sets of terms of a certain type. Our version of candidates for $NRC_\lambda$ is a straightforward adaptation of the standard definition given by Girard and like that one is based on a notion of *neutral terms*, i.e. those terms that, when placed in an arbitrary context, do not create additional redexes. The set of neutral terms is denoted by $\mathcal{NT}$. Let us introduce the following notation for Girard's CRx properties of sets:

- $\mathrm{CR1}(\mathcal{C}) \triangleq \mathcal{C} \subseteq \mathcal{SN}$
- $\mathrm{CR2}(\mathcal{C}) \triangleq \forall M \in \mathcal{C}, M'.M \rightsquigarrow M' \Longrightarrow M' \in \mathcal{C}$
- $\mathrm{CR3}(\mathcal{C}) \triangleq \forall M \in \mathcal{NT}.(\forall M'.M \rightsquigarrow M' \Longrightarrow M' \in \mathcal{C}) \Longrightarrow M \in \mathcal{C}$

The set $\mathcal{CR}$ of the candidates of reducibility is then defined as the collection of those sets of terms which satisfy all the CRx properties. Some standard results include the non-emptiness of candidates (in particular, all free variables are in every candidate) and that $\mathcal{SN} \in \mathcal{CR}$.

## 3.4 Reducibility sets

In this section we introduce *reducibility sets*, which are sets of terms that we will use to provide an interpretation of the types of $NRC_\lambda$; we will then prove that reducibility sets are candidates of reducibility, hence they only contain strongly normalizing terms. The following notation will be useful as a shorthand for certain operations on sets of terms that are used to define reducibility sets:

- $\mathcal{C} \to \mathcal{D} \triangleq \{M : \forall N \in \mathcal{C}, (M\ N) \in \mathcal{D}\}$
- $\langle \overrightarrow{\ell_k : \mathcal{C}_k} \rangle \triangleq \{M : \forall i = 1, \ldots, k, M.\ell_i \in \mathcal{C}_i\}$
- $(p : \mathcal{C})^\top \triangleq \{K : \forall M \in \mathcal{C}.K[p \mapsto \{M\}] \in \mathcal{SN}\}$
- $\mathcal{C}^{\top\top} \triangleq \{M : \forall p, \forall K \in (p : \mathcal{C})^\top, K[p \mapsto M] \in \mathcal{SN}\}$

The sets $(p : \mathcal{C})^\top$ and $\mathcal{C}^{\top\top}$ are called the $\top$-lifting and $\top\top$-lifting of $\mathcal{C}$. These definitions refine the ones used in the literature by using indices: $\top$-lifting is defined with respect to a given index $p$, while the definition of $\top\top$-lifting uses any index.

▶ **Definition 18** (reducibility). *For all types $T$, the set $\mathsf{Red}_T$ of reducible terms of type $T$ is defined by recursion on $T$ by means of the rules:*

$$\mathsf{Red}_A \triangleq \mathcal{SN} \qquad \mathsf{Red}_{S \to T} \triangleq \mathsf{Red}_S \to \mathsf{Red}_T \qquad \mathsf{Red}_{\langle \overrightarrow{\ell_k : T_k} \rangle} \triangleq \langle \overrightarrow{\ell_k : \mathsf{Red}_{T_k}} \rangle \qquad \mathsf{Red}_{\{T\}} \triangleq \mathsf{Red}_T^{\top\top}$$

Let us use metavariables $\Theta, \Theta', \ldots$ to denote finite maps from indices to sets of terms in the form $(p_1 : \mathcal{C}_1, \ldots, p_k : \mathcal{C}_k)$. We extend the notion of $\top$-lifting to such maps by taking the intersection of all the $(p_i : \mathcal{C}_i)^\top$. This notation is useful to track $\Theta$ under renaming reduction.

▶ **Definition 19.** $\Theta^\top \triangleq \bigcap_{p \in \mathrm{dom}(\Theta)} (p : \Theta(p))^\top$

▶ **Definition 20.** *Let $\Theta$ be a finite map from indices to sets of terms and $\sigma$ a renaming: then we define $\Theta^\sigma$ as the finite map $\Theta^\sigma(p) = \Theta(\sigma(p))$, defined for all $p$ such that $\sigma(p) \in \mathrm{dom}(\Theta)$.*

We now proceed with the proof that all the sets $\mathsf{Red}_T$ are candidates of reducibility: we will only focus on collections since for the other types the result is standard. The proofs of CR1 and CR2 do not differ much from the standard $\top\top$-lifting technique.

▶ **Lemma 21** (CR1 for continuations). *For all $p$ and all non-empty $\mathcal{C}$, $(p : \mathcal{C})^\top \subseteq \mathcal{SN}$.*

▶ **Lemma 22** (CR1 for collections). *Suppose* $\mathrm{CR1}(\mathcal{C})$*: then* $\mathrm{CR1}(\mathcal{C}^{\top\top})$.

▶ **Lemma 23** (CR2 for collections). *Suppose $M \in \mathcal{C}^{\top\top}$, and $M \rightsquigarrow M'$: then $M' \in \mathcal{C}^{\top\top}$.*

In order to prove CR2 for all types (and particularly for collections), we do not need to establish an analogous property on continuations; however such a property is still useful for subsequent results (particularly CR3): its statement must, of course, consider that reduction may duplicate (or indeed delete) holes, and thus employs renaming reduction.

▶ **Lemma 24** (CR2 for continuations). *If $K \in \Theta^\top$ and $K \overset{\sigma}{\rightsquigarrow} K'$, then $K' \in (\Theta^\sigma)^\top$.*

The lemma above could have some rather scary consequences for our proof: since reducing a term-in-continuation can lead to duplication, every proof of a statement about the strong normalizability of a term-in-continuation that proceeds by induction on its reduction chains would need to be generalized to $n$-ary instantiations of $n$-ary continuations! Fortunately, there is a better solution: instantiations to pure terms are always permutable, therefore we can simply consider each hole separately, as stated in the following lemma.

▶ **Lemma 25.** *$K \in (\Theta^\sigma)^\top$ if, and only if, for all $q \in \mathrm{dom}(\Theta^\sigma)$, we have $K \in (q : \Theta(\sigma(q)))^\top$. In particular, $K \in ((p : \mathcal{C})^\sigma)^\top$ if, and only if, for all $q$ s.t. $\sigma(q) = p$, we have $K \in (q : \mathcal{C})^\top$.*

This is everything we need to prove CR3.

▶ **Lemma 26** (CR3 for collections). *Let $\mathcal{C} \in \mathcal{CR}$, and $M$ a neutral term such that for all reductions $M \rightsquigarrow M'$ we have $M' \in \mathcal{C}^{\top\top}$. Then $M \in \mathcal{C}^{\top\top}$.*

**Proof.** By definition, we need to prove $K[p \mapsto M] \in \mathcal{SN}$ whenever $K \in (p : \mathcal{C})^\top$ for some index $p$. By Lemma 21, knowing that $\mathcal{C}$, being a candidate, is non-empty, we have $K \in \mathcal{SN}$. We can thus proceed by well-founded induction on $\nu(K)$ to prove the strengthened statement: for all indices $q$, if $K \in (q : \mathcal{C})^\top$, then $K[q \mapsto M] \in \mathcal{SN}$. Equivalently, we prove that all the contracta of $K[q \mapsto M]$ are s.n. by cases on the possible contracta:

- $K'[q \mapsto M]^\sigma$ (where $K \overset{\sigma}{\rightsquigarrow} K'$): to prove this term is s.n., by Lemma 17, we need to prove $K'[q' \mapsto M] \in \mathcal{SN}$ whenever $\sigma(q') = q$; by Lemma 24 and 25, we know $K' \in (q' : \mathcal{C})^\top$, and naturally $\nu(K') < \nu(K)$ (Lemma 10), thus the thesis follows by the IH.
- $K[p \mapsto M']$ (where $M \rightsquigarrow M'$): this is s.n. because $M' \in \mathcal{C}^{\top\top}$ by hypothesis.
- Since $M$ is neutral, there are no reductions at the interface. ◀

▶ **Theorem 27.** *For all types $T$, $\mathsf{Red}_T \in \mathcal{CR}$.*

**Proof.** Standard by induction on $T$. For $T = \{T'\}$, we use Lemma 22, 23, and 26. ◀

## 4 Strong normalization

Having proved that the reducibility sets of all types are candidates of reducibility, in order to prove strong normalization we only need to know that every well-typed term is in the reducibility set corresponding to its type: this proof is by structural induction on the derivation of the typing judgment. Reducibility of singletons is trivial by definition, while that of empty collections is proved in the same style as [6], with the obvious adaptations.

▶ **Lemma 28** (reducibility for singletons). *For all $\mathcal{C}$, if $M \in \mathcal{C}$, then $\{M\} \in \mathcal{C}^{\top\top}$.*

▶ **Lemma 29** (reducibility for $\emptyset$). *For all $\mathcal{C}$, $\emptyset \in \mathcal{C}^{\top\top}$.*

As for unions, we will prove a more general statement on auxiliary continuations.

▶ **Lemma 30.**
*For all auxiliary continuations $Q, O_1, O_2$ with pairwise disjoint supports, if $Q[p \mapsto O_1] \in \mathcal{SN}$ and $Q[p \mapsto O_2] \in \mathcal{SN}$, then $Q[p \mapsto O_1 \cup O_2] \in \mathcal{SN}$.*

**Proof sketch.** The proof follows the same style as [6]; however since our definition of auxiliary continuations is more general than his, the theorem statement mentions $O_1, O_2$ rather than pure terms: the hypothesis on the supports of the continuations being disjoint is required by this generalization. ◀

▶ **Corollary 31** (reducibility for unions). *If $M \in \mathcal{C}^{\top\top}$ and $N \in \mathcal{C}^{\top\top}$, then $M \cup N \in \mathcal{C}^{\top\top}$.*

Like in proofs based on standard $\top\top$-lifting, the most challenging cases are those dealing with commuting conversions – in our case, comprehensions and conditionals.

▶ **Lemma 32.** *Let $K$, $\overline{L}$, $\overline{N}$ such that $K[p \mapsto \overline{N}\left[\overline{L}/x\right]] \in \mathcal{SN}$ and $\overline{L} \in \mathcal{SN}$. Then, $K[p \mapsto \bigcup\{\overline{N}|x \leftarrow \{\overline{L}\}\}] \in \mathcal{SN}$.*

**Proof.** In this proof, we assume the names of bound variables are chosen so as to avoid duplicates, and distinct from the free variables. We proceed by well-founded induction on $(K, p, \overline{N}, \overline{L})$ using the following metric:

$$\begin{aligned}&(K_1, p_1, \overline{N_1}, \overline{L_1}) \prec (K_2, p_2, \overline{N_2}, \overline{L_2})\\ &\iff (\nu(K_1[p_1 \mapsto \overline{N_1}\left[\overline{L_1}/x\right]]) + \nu(\overline{L_1}), \|K_1\|_{p_1}, \text{size}(\overline{N_1}))\\ &\qquad \lessdot (\nu(K_2[p_2 \mapsto \overline{N_2}\left[\overline{L_2}/x\right]]) + \nu(\overline{L_2}), \|K_2\|_{p_2}, \text{size}(\overline{N_2}))\end{aligned}$$

Now we show that every contractum must be a strongly normalizing:

- $K[p \mapsto \overline{N}\left[\overline{L}/x\right]]$: this term is s.n. by hypothesis.
- $K'[p \mapsto \bigcup\{N|x \leftarrow \{\overline{L}\}\}]^\sigma$, where $K \overset{\sigma}{\rightsquigarrow} K'$. By Lemma 10 we know $\nu(K'[p \mapsto \overline{N}\left[\overline{L}/x\right]]^\sigma) < \nu(K[p \mapsto \overline{N}\left[\overline{L}/x\right]])$ (since the former is a contractum of the latter), which implies $\nu(K'[q \mapsto \overline{N}\left[\overline{L}/x\right]]) \leq \nu(K'[p \mapsto \overline{N}\left[\overline{L}/x\right]]^\sigma) < \nu(K[p \mapsto \overline{N}\left[\overline{L}/x\right]])$ for all $q$ s.t. $\sigma(q) = p$ by means of Lemma 54 (because $[q \mapsto \overline{N}\left[\overline{L}/x\right]]$ is a subapplication of $[p \mapsto \overline{N}\left[\overline{L}/x\right]]^\sigma$); then we can apply the IH to obtain, for all $q$ s.t. $\sigma(q) = p$, $K'[q \mapsto \bigcup\{\overline{N}|x \leftarrow \{\overline{L}\}\}] \in \mathcal{SN}$; by Lemma 17, this implies the thesis.
- $K[p \mapsto \emptyset]$ (when $N = \emptyset$): this is equal to $K[p \mapsto \emptyset\left[\overline{L}/x\right]]$, which is s.n. by hypothesis.
- $K[p \mapsto \bigcup\{\overline{N_1}|x \leftarrow \{\overline{L}\}\} \cup \bigcup\{\overline{N_2}|x \leftarrow \{\overline{L}\}\}]$ (when $\overline{N} = \overline{N_1} \cup \overline{N_2}$); by IH (since $\text{size}(\overline{N_i}) < \text{size}(\overline{N_1} \cup \overline{N_2})$, and all other metrics do not increase) we prove $K[p \mapsto \bigcup\{\overline{N_i}|x \leftarrow \{\overline{L}\}\}]$ (for $i = 1, 2$) by IH, and consequently obtain the thesis by Lemma 30.
- $K_0[p \mapsto \bigcup\{\bigcup\{\overline{M}|y \leftarrow \overline{N}\}|x \leftarrow \{\overline{L}\}\}]$, where $K = K_0 \circledcirc_p \bigcup\{\overline{M}|y\}$; since we know, by the hypothesis on the choice of bound variables, that $x \notin \text{FV}(\overline{M})$, we note that $K_0[p \mapsto \bigcup\{\overline{M}|y \leftarrow \overline{N}\}\left[\overline{L}/x\right]] = K[p \mapsto \overline{N}\left[\overline{L}/x\right]]$; furthermore, we know $\|K_0\|_p < \|K\|_p$; then we can apply the IH to obtain the thesis.
- $K_0[p \mapsto \bigcup\{\texttt{where}\ \overline{B}\ \texttt{do}\ \overline{N}|x \leftarrow \{\overline{L}\}\}]$ (when $K = K_0 \circledcirc_p \texttt{where}\ \overline{B}$): since we know, from the hypothesis on the choice of bound variables, that $x \notin \text{FV}(B)$, we note that $K_0[p \mapsto (\texttt{where}\ \overline{B}\ \texttt{do}\ \overline{N})\left[\overline{L}/x\right]] = K[p \mapsto \overline{N}\left[\overline{L}/x\right]]$; furthermore, we know $\|K_0\|_p < \|K\|_p$; then we can apply the IH to obtain the thesis.
- reductions within $N$ or $L$ follow from the IH by reducing the induction metric. ◀

▶ **Lemma 33** (reducibility for comprehensions). *Assume CR1($\mathcal{C}$), CR1($\mathcal{D}$), $\overline{M} \in \mathcal{C}^{\top\top}$ and for all $\overline{L} \in \mathcal{C}$, $\overline{N}\left[\overline{L}/x\right] \in \mathcal{D}^{\top\top}$. Then $\bigcup\{\overline{N}|x \leftarrow \overline{M}\} \in \mathcal{D}^{\top\top}$.*

**Proof.** We assume $p$, $K \in (p : \mathcal{D})^\top$ and prove $K[p \mapsto \bigcup\{\overline{N}|x \leftarrow \overline{M}\}] \in \mathcal{SN}$. We start by showing that $K' = K \circledcirc_p \bigcup\{\overline{N}|x\} \in (p : \mathcal{C})^\top$, or equivalently that for all $\overline{L} \in \mathcal{C}$, $K'[p \mapsto \{\overline{L}\}] = K[p \mapsto \bigcup\{\overline{N}|x \leftarrow \{\overline{L}\}\}] \in \mathcal{SN}$: since CR1($\mathcal{C}$), we know $\overline{L} \in \mathcal{SN}$, and since $\overline{N}\left[\overline{L}/x\right] \in \mathcal{D}^{\top\top}$, $K[p \mapsto \overline{N}\left[\overline{L}/x\right]] \in \mathcal{SN}$; then we can apply Lemma 32 to obtain $K'[p \mapsto \{\overline{L}\}] \in \mathcal{SN}$ and consequently $K' \in (p : \mathcal{C})^\top$. But then, since $\overline{M} \in \mathcal{C}^{\top\top}$, we have $K'[p \mapsto \overline{M}] = K[p \mapsto \bigcup\{\overline{N}|x \leftarrow \overline{M}\}] \in \mathcal{SN}$, which is what we needed to prove. ◀

Reducibility for conditionals is proved in a similar manner. However, to consider all the conversions commuting with `where`, we need to use the more general auxiliary continuations.

▶ **Lemma 34.** *Let $Q$, $\overline{B}$, $O$ such that $Q[p \mapsto O] \in \mathcal{SN}$, $\overline{B} \in \mathcal{SN}$, and $\mathsf{supp}(Q) \cap \mathsf{supp}(O) = \emptyset$. Then $Q[p \mapsto \texttt{where}\ \overline{B}\ \texttt{do}\ O] \in \mathcal{SN}$.*

**Proof sketch.** We proceed by well-founded induction on $(Q, B, O, p)$ using the following metric:

$$(Q_1, \overline{B_1}, O_1, p_1) \prec (Q_2, \overline{B_2}, O_2, p_2) \iff \\ (\nu(Q_1[p_1 \mapsto O_1]) + \nu(\overline{B_1}), |Q_1|_{p_1}, \mathrm{size}(O_1)) \lessdot (\nu(Q_2[p_2 \mapsto O_2]) + \nu(\overline{B_2}), |Q_2|_{p_2}, \mathrm{size}(O_2))$$

We show every contractum must be a strongly normalizing term; we apply the IH to new auxiliary continuations obtained by placing pieces of $O$ into $Q$ or vice-versa: the hypothesis on the supports of $Q$ and $O$ is used to ensure that the new continuations are well-formed. The use of $|\cdot|_p$ rather than $\|\cdot\|_p$ is needed to ensure that contractions in the form $Q[p \mapsto \texttt{where}\ \overline{B}\ \texttt{do}\ \bigcup\{O_1|x \leftarrow O_2\}] \rightsquigarrow (Q \circledcirc_p \bigcup\{x \leftarrow O_2\})[p \mapsto \texttt{where}\ \overline{B}\ \texttt{do}\ O_1]$ do not increase the metric. ◀

▶ **Corollary 35** (reducibility for conditionals). *If $\overline{B} \in \mathcal{SN}$ and $\overline{N} \in \mathsf{Red}_{\{T\}}$, then $\texttt{where}\ \overline{B}\ \texttt{do}\ \overline{N} \in \mathsf{Red}_{\{T\}}$.*

Finally, reducibility for the emptiness test is proved in the same style as [6].

▶ **Lemma 36.** *For all $M$ and $T$ such that $\Gamma \vdash M : \{T\}$ and $M \in \mathsf{Red}_T^{\top\top}$, we have $\texttt{empty}(M) \in \mathcal{SN}$.*

## 4.1 Main theorem

Before stating and proving the main theorem, we introduce some auxiliary notation.

▶ **Definition 37.**

1. *A substitution $\rho$ satisfies $\Gamma$ (notation: $\rho \vDash \Gamma$) iff, for all $x \in \mathrm{dom}(\Gamma)$, $\rho(x) \in \mathsf{Red}_{\Gamma(x)}$.*
2. *A substitution $\rho$ satisfies $M$ with type $T$ (notation: $\rho \vDash M : T$) iff $M\rho \in \mathsf{Red}_T$.*

As usual, the main result is obtained as a corollary of a stronger theorem generalized to substitutions into open terms, by using the identity substitution $\mathsf{id}_\Gamma$.

▶ **Lemma 38.** *For all $\Gamma$, we have $\mathsf{id}_\Gamma \vDash \Gamma$.*

▶ **Theorem 39.** *If $\Gamma \vdash M : T$, then for all $\rho$ such that $\rho \vDash \Gamma$, we have $\rho \vDash M : T$*

**Proof.** By induction on the derivation of $\Gamma \vdash M : T$. When $M$ is empty, a singleton, a union, an emptiness test, or a conditional, we use Lemma 29, 28, 31, 36, and 35. For comprehensions such that $\Gamma \vdash \bigcup\{M_1|x \leftarrow M_2\} : \{T\}$, we know by IH that $\rho \vDash M_2 : \{S\}$ and for all $\rho' \vDash \Gamma, x : S$ we have $\rho' \vDash M_1 : \{T\}$: we prove that for all $L \in \mathsf{Red}_S$, $\rho\,[L/x] \vDash \Gamma, x : S$, hence $\rho\,[L/x] \vdash M_1 : \{T\}$; then we obtain $\rho \vDash \bigcup\{M_1|x \leftarrow M_2\} : \{T\}$ by Lemma 33. Non-collection cases are standard. ◀

▶ **Corollary 40.** *If $\Gamma \vdash M : T$, then $M \in \mathcal{SN}$.*

# 5 Heterogeneous Collections

In a short paper [19], we introduced a generalization of *NRC* called *NRC*(*Set*, *Bag*), which contains both set-valued and bag-valued collections (with distinct types denoted by $\{T\}$ and $\wr T \wr$), along with mapping from bags to sets (deduplication $\delta$) and from sets to bags (promotion $\iota$). We conjectured that this language also satisfies a normalization property. Here, we prove this claim, even extending $NRC(Set, Bag)$ to a richer langauage $NRC_\lambda(Set, Bag)$ with higher-order (nonrecursive) functions.

$$\begin{array}{rcl} L, M, N & ::= & \ldots \mid \mho \mid \wr M \wr \mid M \uplus N \mid \biguplus \wr M | x \leftarrow N \wr \\ & \mid & \mathtt{where}_{\mathsf{bag}}\ M\ \mathtt{do}\ N \mid \mathtt{empty}_{\mathsf{bag}}\ M \mid \delta M \mid \iota M \end{array}$$

The notations $\mho$, $\wr M \wr$, $M \uplus N$, $\biguplus \wr M | x \leftarrow N \wr$ denote empty and singleton bags, bag disjoint union and bag comprehension; the language also includes conditionals and emptiness tests on bags. We omit the typing rules, and observe that the reduction rules involving bag operations correspond to those for set operations, and additionally include the following:

$$\begin{array}{c} \delta\mho \rightsquigarrow \emptyset \qquad \delta \wr M \wr \rightsquigarrow \{M\} \qquad \delta(M \uplus N) \rightsquigarrow \delta M \cup \delta N \qquad \delta\iota M \rightsquigarrow M \\ \delta \biguplus \wr M | x \leftarrow N \wr \rightsquigarrow \bigcup \{\delta M | x \leftarrow \delta N\} \qquad \delta(\mathtt{where}_{\mathsf{bag}}\ M\ \mathtt{do}\ N) \rightsquigarrow \mathtt{where}\ M\ \mathtt{do}\ \delta N \\ \iota\emptyset \rightsquigarrow \mho \qquad \iota\{M\} \rightsquigarrow \wr M \wr \qquad \iota(\mathtt{where}\ M\ \mathtt{do}\ N) \rightsquigarrow \mathtt{where}_{\mathsf{bag}}\ M\ \mathtt{do}\ \iota N \end{array}$$

SN for $NRC_\lambda(Set, Bag)$ is proved by first translating the language to a version of $NRC_\lambda$ retaining the operations $\delta$ and $\iota$ that we call $NRC_{\lambda\delta\iota}$, by means of a forgetful translation $\lfloor \cdot \rfloor$ mapping empty bags, bag unions and bag comprehensions to the corresponding set constructs. We prove that every contraction in $NRC_\lambda(Set, Bag)$ is translated to a contraction in $NRC_{\lambda\delta\iota}$, and thus obtain SN for $NRC_\lambda(Set, Bag)$ as a corollary of SN for $NRC_{\lambda\delta\iota}$.

▶ **Theorem 41.** *If $\Gamma \vdash M : T$ in $NRC_\lambda(Set, Bag)$, then $\lfloor \Gamma \rfloor \vdash \lfloor M \rfloor : \lfloor T \rfloor$ in $NRC_{\lambda\delta\iota}$.*

▶ **Lemma 42.** *For all terms $M$ of $NRC_\lambda(Set, Bag)$, if $M \rightsquigarrow M'$, we have $\lfloor M \rfloor \rightsquigarrow \lfloor M' \rfloor$ in $NRC_{\lambda\delta\iota}$. Consequently, if $\lfloor M' \rfloor \in \mathcal{SN}$ in $NRC_{\lambda\delta\iota}$, then $M' \in \mathcal{SN}$ in $NRC_\lambda(Set, Bag)$.*

▶ **Theorem 43.** *If $\Gamma \vdash M : T$ in $NRC_{\lambda\delta\iota}$, then $M \in \mathcal{SN}$ in $NRC_{\lambda\delta\iota}$.*

▶ **Corollary 44.** *If $\Gamma \vdash M : T$ in $NRC_\lambda(Set, Bag)$, then $M \in \mathcal{SN}$ in $NRC_\lambda(Set, Bag)$.*

## 6 Related Work

This paper builds on a long line of research on normalisation of comprehension queries, a model of query languages popularized over 25 years ago by Buneman et al. [2]. Wong [20] proved conservativity via a strongly normalising rewrite system, which was used in Kleisli [21], a functional query system, in which flat query expressions were normalised to SQL. Libkin and Wong [12, 13] investigated conservativity in the presence of aggregates, internal generic functions, and bag operations, and demonstrated that bag operations can be expressed using nested comprehensions. However, their normalization results studied bag queries by translating to relational queries with aggregation, and did not consider higher-order queries, so they do not imply the normalization results for $NRC_\lambda(Set, Bag)$ given here.

Cooper [7] first investigated query normalisation (and hence conservativity) in the presence of higher-order functions. He gave a rewrite system showing how to normalise homogeneous (that is, pure set or pure bag) queries to eliminate intermediate occurrences of nesting or of function types. However, although Cooper claimed a proof (based on ⊤⊤-lifting [15]) and provided proof details in his PhD thesis [6], there unfortunately turned out to be a nontrivial lacuna in that proof, and this paper therefore (in our opinion) contains the first *complete* proof of normalisation for higher-order queries, even for the homogeneous case.

Since the fundamental work of Wong and others on the Kleisli system, language-integrated query has gradually made its way into other systems, most notably Microsoft's .NET framework languages C# and F# [16], and the Web programming language Links [8]. Cheney et al. [3] formally investigated the F# approach to language-integrated query and showed that normalisation results due to Wong and Cooper could be adapted to improve it further; however, their work considered only homogeneous collections. In subsequent work, Cheney et al. [4] showed how use normalisation to perform *query shredding* for multiset queries, in which a query returning a type with $n$ nested collections can be implemented by combining the results of $n$ flat queries; this has been implemented in Links [8].

Seeral recent efforts to formalize and reason about the semantics of SQL are complementary to our work. Guagliardo and Libkin [10] presented a semantics for SQL's actual behaviour in the presence of set and multiset operators (including bag intersection and difference) as well as incomplete information (nulls), and related the expressiveness of this fragment of SQL with that of an algebra over bags with nulls. Chu et al. [5] presented a formalised semantics for reasoning about SQL (including set and bag semantics as well as aggregation/grouping, but excluding nulls) using nested relational queries in Coq, while Benzaken and Contejean [1] presented a semantics including all of these SQL features (set, multiset, aggregation/grouping, nulls), and formalised the semantics in Coq. Kiselyov et al. [11] has proposed language-integrated query techniques that handle sorting operations (SQL's `ORDER BY`).

However, the above work on semantics has not considered query normalisation, and to the best of our knowledge normalisation results for query languages with more than one collection type were previously unknown even in the first-order case. We are interested in extending our results for mixed set and bag semantics to handle nulls, grouping/aggregation, and sorting, thus extending higher-order language integrated query to cover all of the most widely-used SQL features. To the best of our knowledge, normalisation of higher-order queries in the presence of all of these features simultaneously remains an open problem, which we plan to consider next. In addition, fully formalising such normalisation proofs also appears to be a nontrivial challenge.

## 7 Conclusions

Integrating database queries into programming languages has many benefits, such as type safety and avoidance of common SQL injection attacks, but also imposes limitations that prevent programmers from constructing queries dynamically as they could by concatenating SQL strings unsafely. Previous work has demonstrated that many useful dynamic queries can be constructed safely using *higher-order functions* inside language-integrated queries; provided such functions are not recursive, it was believed, query expressions can be normalised. Moreover, while it is common in practice to provide support for SQL features such as mixed set and bag operators, it is not well understood in theory how to normalise these queries in the presence of higher-order functions. Previous work on higher-order query normalisation has considered only homogeneous (that is, pure set or pure bag) queries, and in the process of attempting to generalise this work to a heterogeneous setting, we discovered a nontrivial gap in the previous proof of strong normalisation. We therefore prove strong normalisation for both homogeneous and heterogeneous queries for the first time.

As next steps, we intend to extend the Links implementation of language-integrated query with heterogeneous queries and normalisation, and to investigate (higher-order) query normalisation and conservativity for the remaining common SQL features, such as nulls, grouping/aggregation, and ordering.

## References

1 Véronique Benzaken and Evelyne Contejean. A Coq mechanised formal semantics for realistic SQL queries: formally reconciling SQL and bag relational algebra. In *CPP 2019*, pages

249–261, 2019.

2 Peter Buneman, Shamim Naqvi, Val Tannen, and Limsoon Wong. Principles of programming with complex objects and collection types. *Theor. Comput. Sci.*, 149(1), 1995.

3 James Cheney, Sam Lindley, and Philip Wadler. A practical theory of language-integrated query. In *ICFP*, 2013.

4 James Cheney, Sam Lindley, and Philip Wadler. Query shredding: efficient relational evaluation of queries over nested multisets. In *SIGMOD*, pages 1027–1038. ACM, 2014.

5 Shumo Chu, Konstantin Weitz, Alvin Cheung, and Dan Suciu. HoTTSQL: Proving query rewrites with univalent SQL semantics. In *PLDI*, pages 510–524. ACM, 2017.

6 Ezra Cooper. *Programming language features for web application development.* PhD thesis, University of Edinburgh, 2009.

7 Ezra Cooper. The script-writer's dream: How to write great SQL in your own language, and be sure it will succeed. In *DBPL*, 2009.

8 Ezra Cooper, Sam Lindley, Philip Wadler, and Jeremy Yallop. Links: web programming without tiers. In *FMCO*, 2007.

9 Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types.* Cambridge University Press, 1989.

10 Paolo Guagliardo and Leonid Libkin. A formal semantics of SQL queries, its validation, and applications. *PVLDB*, 2017.

11 Oleg Kiselyov and Tatsuya Katsushima. Sound and efficient language-integrated query - maintaining the ORDER. In *APLAS 2017*, pages 364–383, 2017.

12 Leonid Libkin and Limsoon Wong. Conservativity of nested relational calculi with internal generic functions. *Inf. Process. Lett.*, 49(6):273–280, 1994.

13 Leonid Libkin and Limsoon Wong. Query languages for bags and aggregate functions. *J. Comput. Syst. Sci.*, 55(2), 1997.

14 Sam Lindley and James Cheney. Row-based effect types for database integration. In *TLDI*, 2012.

15 Sam Lindley and Ian Stark. Reducibility and $\top\top$-lifting for computation types. In *TLCA*, 2005.

16 Erik Meijer, Brian Beckman, and Gavin M. Bierman. LINQ: reconciling object, relations and XML in the .NET framework. In *SIGMOD*, 2006.

17 Jan Paredaens and Dirk Van Gucht. Converting nested algebra expressions into flat algebra expressions. *ACM Trans. Database Syst.*, 17(1), 1992.

18 W. Ricciotti and J. Cheney. Strongly Normalizing Audited Computation. In V. Goranko and M. Dam, editors, *CSL 2017*, volume 82 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

19 Wilmer Ricciotti and James Cheney. Mixing set and bag semantics. In *DBPL*, pages 70–73, 2019.

20 Limsoon Wong. Normal forms and conservative extension properties for query languages over collection types. *J. Comput. Syst. Sci.*, 52(3), 1996.

21 Limsoon Wong. Kleisli, a functional query system. *J. Funct. Programming*, 10(1), 2000.

# A Proofs

This appendix expands on some results whose proofs were omitted or only sketched in the paper.

Since under plain reduction each term can be reduced only in a finite number of ways, it is easy to see that $\nu(M)$ is defined for any strongly normalizing term $M$; however, under renaming reduction, a term may be reduced in an infinite number of ways because, if $M \rightsquigarrow N$, there may be infinite $R, \sigma$ such that $N = R\sigma$. Fortunately, we can prove that to any renaming reduction chain there corresponds a plain reduction chain of the same length, and vice-versa: consequently, the set of strongly normalizing terms is the same under the two notions of reduction, and $\nu(M)$ refers to the maximal length of reduction chains starting at $M$ either with or without renaming.

▶ **Lemma 45.** *For all contexts $C$, terms $N$ and indices $p$, if $C[p \mapsto N] \in \mathcal{SN}$, we have $C \in \mathcal{SN}$; if $p \in \mathsf{supp}(C)$, then $N \in \mathcal{SN}$.*

▶ **Lemma 46.** *If $M \rightsquigarrow N$, then $M\sigma \rightsquigarrow N\sigma$.*

▶ **Lemma 47.**

1. *If $M \underbrace{\rightsquigarrow \cdots \rightsquigarrow}_{n \text{ times}} N$, then $M \underbrace{\overset{\mathsf{id}}{\rightsquigarrow} \cdots \overset{\mathsf{id}}{\rightsquigarrow}}_{n \text{ times}} N$*
2. *If $M \overset{\sigma_1}{\rightsquigarrow} \cdots \overset{\sigma_n}{\rightsquigarrow} N$, then $M \underbrace{\rightsquigarrow \cdots \rightsquigarrow}_{n \text{ times}} N\sigma_n \cdots \sigma_1$*

**Proof.** The first part of the lemma is trivial. For the second part, proceed by induction on the length of the reduction chain: in the inductive case, we have $M \overset{\sigma_1}{\rightsquigarrow} \cdots \overset{\sigma_n}{\rightsquigarrow} M' \overset{\sigma_{n+1}}{\rightsquigarrow} N$ by hypothesis and $M \rightsquigarrow \cdots \rightsquigarrow M'\sigma_n \cdots \sigma_1$ by induction hypothesis; to obtain the thesis, we only need to prove that

$$M'\sigma_n \cdots \sigma_1 \rightsquigarrow N\sigma_{n+1} \cdots \sigma_1$$

In order for this to be true, by Lemma 46, it is sufficient to show that $M' \rightsquigarrow N\sigma_{n+1}$; this is by definition equivalent to $M' \overset{\sigma_{n+1}}{\rightsquigarrow} N$, which we know by hypothesis. ◀

▶ **Corollary 48.** *Suppose $M \in \mathcal{SN}$: if $M \overset{\sigma}{\rightsquigarrow} M'$, then $\nu(M')$ is defined and $\nu(M') < \nu(M)$.*

**Proof.** By Lemma 47, for any plain reduction chain there exists a renaming reduction chain of the same length, and vice-versa. Thus, since plain reduction lowers the length of the maximal reduction chain (Lemma 10), the same holds for renaming reduction. ◀

**Proof of Lemma 13.**

1. *For all continuations $K$, if $K \rightsquigarrow C$, there exist a continuation $K'$ and a finite map $\sigma$ such that $K \overset{\sigma}{\rightsquigarrow} K'$ and $K'\sigma = C$.*
2. *For all auxiliary continuation $Q$, if $Q \rightsquigarrow C$, there exist an auxiliary continuation $Q'$ and a finite map $\sigma$ such that $Q \overset{\sigma}{\rightsquigarrow} Q'$ and $Q'\sigma = C$.*

Let $C$ be a contractum of the continuation we wish to reduce. This contractum will not, in general, satisfy the side condition that holes must be linear; however we can show that, for any context with duplicated holes, there exists a structurally equal context with linear holes. Operationally, if $C$ contains $n$ holes, we generate $n$ different fresh indices in $\mathcal{P}$, and replace the index of each hole in $C$ with a different fresh index to obtain a new context $C'$: this induces a finite map $\sigma : \mathsf{supp}(C') \to \mathsf{supp}(C)$ such that $C'\sigma = C$. By structural induction on the derivation of the reduction and by case analysis on the structure of $K$ (or on the structure of $Q$) we show that $C'$ must also satisfy the grammar in Definition 5 (resp. Definition 6); furthermore, $C'$ satisfies the linearity condition by construction, which proves it is a continuation $K'$ (resp. an auxiliary continuation $Q'$). ◀

▶ **Lemma 49.** *For all contexts $C$ and hole instantiations $\eta$, if $C \rightsquigarrow C'$, then $C\eta \rightsquigarrow C'\eta$.*

▶ **Lemma 50.** *For all contexts $C$, finite maps $\sigma$, and hole instantiations $\eta$ such that, for all $p \in \mathrm{dom}(\eta)$, $\mathsf{supp}(\eta(p)) \cap \mathrm{dom}(\sigma) = \emptyset$, we have $C\sigma\eta = C\eta^\sigma\sigma$.*

**Proof.** By structural induction on $C$. The interesting case is when $C = [p]$. If $\sigma(p) \in \mathrm{dom}(\eta)$, we have $[p]\,\sigma\eta = [\sigma(p)]\,\eta = \eta(\sigma(p)) = \eta(\sigma(p))\sigma = [p]\,\eta^\sigma\sigma$; otherwise, $[p]\,\sigma\eta = [p] = [p]\,\eta^\sigma\sigma$. ◀

**Proof of Lemma 15.** *For all contexts $C$, renamings $\sigma$, and hole instantiations $\eta$ such that, for all $p \in \mathrm{dom}(\eta)$, $\mathsf{supp}(\eta(p)) \cap \mathrm{dom}(\sigma) = \emptyset$, if $C \overset{\sigma}{\rightsquigarrow} C'$, then $C\eta \overset{\sigma}{\rightsquigarrow} C'\eta^\sigma$.*

By definition of $\overset{\sigma}{\rightsquigarrow}$, we have $C \rightsquigarrow C'\sigma$; then, by Lemma 49, we obtain $C\eta \rightsquigarrow C'\sigma\eta$; by Lemma 50, we know $C'\sigma\eta = C'\eta^\sigma\sigma$; then the thesis $C\eta \overset{\sigma}{\rightsquigarrow} C'\eta^\sigma$ follows immediately by the definition of $\overset{\sigma}{\rightsquigarrow}$. ◀

▶ **Lemma 51.** *Suppose $Q \circledcirc_p f \in \mathcal{SN}$: then, $\nu(Q) \leq \nu(Q \circledcirc_p f)$.*

**Proof.** By induction on the possible reduction sequences in $Q$, we show there exists a corresponding reduction sequence with the same length in $Q \circledcirc_p f$. ◀

▶ **Lemma 52.**
*If $M \rightsquigarrow M'$ and $p \in \mathsf{supp}(Q)$, then $Q[p \mapsto M] \overset{\mathsf{id}}{\rightsquigarrow} Q[p \mapsto M']$.*

**Proof.** By induction on the structure of $Q$, we show that for each reduction in the hypothesis, we can construct a corresponding reduction proving the thesis. ◀

▶ **Lemma 53** (classification of reductions in applied continuations)**.** *Suppose $Q\eta \rightsquigarrow N$, where $\eta$ is permutable, and $\mathrm{dom}(\eta) \subseteq \mathsf{supp}(Q)$; then one of the following holds:*

1. *there exist an auxiliary continuation $Q'$ and a finite map $\sigma$ such that $N = Q'\eta^\sigma$, where $\eta^\sigma$ is permutable, and $Q \overset{\sigma}{\rightsquigarrow} Q'$: in this case, we say the reduction is* within $Q$*;*
2. *there exist auxiliary continuations $Q_1, Q_2$, an index $q \in \mathsf{supp}(Q_1)$, a variable $x$, and a term $L$ such that $Q = (Q_1 \circledcirc_q \bigcup \{x \leftarrow \{\overline{L}\}\})[q \mapsto Q_2]$, and $N = Q_1[q \mapsto Q_2\,[\overline{L}/x]]\eta^*$, where we define $\eta^*(p) = \eta(p)\,[\overline{L}/x]$ for all $p \in \mathsf{supp}(Q_2)$, otherwise $\eta^*(p) = \eta(p)$.: this is a reduction within $Q$ too;*
3. *there exists a permutable $\eta'$ such that $N = Q\eta'$ and $\eta \rightsquigarrow \eta'$: in this case we say the reduction is* within $\eta$*;*
4. *there exist an auxiliary continuation $Q_0$, an index $p$ such that $p \in \mathsf{supp}(Q_0)$ and $p \in \mathrm{dom}(\eta)$, an auxiliary frame $f$ and a term $M$ such that $N = Q_0[p \mapsto M]\eta_{\neg p}$, $Q = Q_0 \circledcirc_p f$, and $f^p[p \mapsto \eta(p)] \rightsquigarrow M$: in this case we say the reduction is* at the interface.

*Furthermore, if $Q$ is a regular continuation $K$, then the $Q'$ in case 1 can be chosen to be a regular continuation $K'$, and case 2 cannot happen.*

**Proof.** By induction on $Q$ with a case analysis on the reduction rule applied. ◀

▶ **Lemma 54.** *If $Q\eta \in \mathcal{SN}$, then $Q \in \mathcal{SN}$ and $\nu(Q) \leq \nu(Q\eta)$.*

**Proof.** We proceed by well-founded induction on $(Q, \eta)$ using the metric:

$$(Q_1, \eta_1) \prec (Q_2, \eta_2) \iff \exists \sigma : Q\eta_1 \overset{\sigma}{\rightsquigarrow} Q'\eta_2$$

For all contractions $Q \overset{\sigma}{\rightsquigarrow} Q'$, by Lemma 52 we know $Q\eta \overset{\sigma}{\rightsquigarrow} Q'\eta^\sigma$: then we can apply the IH with $(Q', \eta^\sigma)$ to prove $Q'$: thus we conclude $Q \in \mathcal{SN}$.

To prove $\nu(Q) \leq \nu(Q\eta)$, it is sufficient to see that for each reduction step in $Q$ we have a corresponding reduction step in $Q\eta$: thus the reduction chains starting in $Q\eta$ must be at least as long as those in $Q$. ◀

▶ **Lemma 55.** *Suppose* $\mathrm{CR1}(\mathcal{C})$*: then for all indices* $p, q$*,* $[p] \in (q \mapsto \mathcal{C})^\top$*.*

**Proof.** To prove the lemma, it is sufficient to show that for all $M \in \mathcal{C}$ we have $[p][q \mapsto \{M\}] \in \mathcal{SN}$. This term is equal to either $\{M\}$ (if $p = q$) or to $[p]$ (otherwise); both terms are s.n. (in the case of $\{M\}$, this is because CR1 holds for $\mathcal{C}$, thus $M \in \mathcal{SN}$). ◀

**Proof of Lemma 21.** *For all* $p$ *and all non-empty* $\mathcal{C}$*,* $(p : \mathcal{C})^\top \subseteq \mathcal{SN}$*.*

We assume $K \in (p : \mathcal{C})^\top$ and $M \in \mathcal{C}$: by definition, we know that $K[p \mapsto \{M\}] \in \mathcal{SN}$; then we have $K \in \mathcal{SN}$ by Lemma 45. ◀

**Proof of Lemma 22.** *Suppose* $\mathrm{CR1}(\mathcal{C})$*: then* $\mathrm{CR1}(\mathcal{C}^{\top\top})$*.*

We need to prove that if $M \in \mathcal{C}^{\top\top}$, then $M \in \mathcal{SN}$. By the definition of $\mathcal{C}^{\top\top}$, we know that for all $p$, $K[p \mapsto M] \in \mathcal{SN}$ whenever $K \in (p : \mathcal{C})^\top$. Now assume any $p$, and by Lemma 55 choose $K = [p]$: then $K[p \mapsto M] = M \in \mathcal{SN}$, which proves the thesis. ◀

▶ **Lemma 56.** *If* $K \in \mathcal{SN}$ *is a continuation, then for all indices* $p$ *we have* $K[p \mapsto \emptyset] \in \mathcal{SN}$*.*

**Proof.** We proceed by well-founded induction, using the metric:

$$(K_1, p_1) \prec (K_2, p_2) \iff (\nu(K_1), \|K_1\|_{p_1}) \lessdot (\nu(K_2), \|K_2\|_{p_2})$$

- $K'[p \mapsto \emptyset]^\sigma$, where $K \overset{\sigma}{\rightsquigarrow} K'$: by Lemma 17, we need to show $K'[q \mapsto \emptyset] \in \mathcal{SN}$ whenever $\sigma(q) = p$; this follows from the IH, with $\nu(K') < \nu(K)$ by Lemma 10.
- $K_0[p \mapsto \emptyset]$, where $K = K_0 \circledcirc_p F$ for some frame $F$: by Lemma 51 we have $\nu(K_0) \leq \nu(K)$; furthermore, we can easily prove that $\|K_0\|_p < \|K\|_p$; then the thesis follows immediately from the IH. ◀

**Proof of Lemma 29.** *For all* $\mathcal{C}$*,* $\emptyset \in \mathcal{C}^{\top\top}$*.*

Immediate from Lemma 56, by unfolding the definition of $\mathcal{C}^{\top\top}$. ◀

**Proof of Lemma 30.** *For all Q-continuations* $Q, O_1, O_2$ *with pairwise disjoint supports, if* $Q[p \mapsto O_1] \in \mathcal{SN}$ *and* $Q[p \mapsto O_2] \in \mathcal{SN}$*, then* $Q[p \mapsto O_1 \cup O_2] \in \mathcal{SN}$*.*

We assume $p \in \mathsf{supp}(Q)$ (otherwise, $Q[p \mapsto O_1] = Q[p \mapsto O_2] = Q[p \mapsto O_1 \cup O_2]$, and the thesis holds trivially). Then, by Lemma 45, $Q[p \mapsto O_1] \in \mathcal{SN}$ and $Q[p \mapsto O_2] \in \mathcal{SN}$ imply $Q \in \mathcal{SN}$, $O_1 \in \mathcal{SN}$, and $O_2 \in \mathcal{SN}$: thus we can proceed by well-founded induction on $(Q, p, O_1, O_2)$ using the following metric:

$$\begin{aligned}&(Q^1, p^1, O_1^1, O_2^1) \prec (Q^2, p^2, O_1^2, O_2^2)\\ &\iff (\nu(Q^1), \|Q^1\|_{p^1}, \nu(O_1^1) + \nu(O_2^1)) \lessdot (\nu(Q^2), \|Q^2\|_{p^2}, \nu(O_1^2) + \nu(O_2^2))\end{aligned}$$

to prove that if $Q[p \mapsto O_1] \in \mathcal{SN}$ and $Q[p \mapsto O_2] \in \mathcal{SN}$, then $Q[p \mapsto O_1 \cup O_2] \in \mathcal{SN}$. Equivalently, we will consider all possible contracta and show that each of them must be a strongly normalizing term; we will apply the induction hypothesis to new auxiliary continuations obtained by placing pieces of $Q$ into $O_1$ and $O_2$: the hypothesis on the supports of the continuations being disjoint is used to make sure that the new continuations do not contain duplicate holes and are thus well-formed. By cases on the possible contracta:

- $Q_1[q \mapsto Q_2 \left[\overline{L}/x\right]][p \mapsto (O_1 \left[\overline{L}/x\right]) \cup (O_2 \left[\overline{L}/x\right])]$ (where $Q = (Q_1 \circledast_q \bigcup \{x \leftarrow \{\overline{L}\}\})[q \mapsto Q_2]$, $q \in \mathsf{supp}(Q_1)$, $p \in \mathsf{supp}(Q_2)$): let $Q' = Q_1[q \mapsto Q_2 \left[\overline{L}/x\right]]$, and note that $Q \rightsquigarrow Q'$, hence $\nu(Q') < \nu(Q)$; note $Q[p \mapsto O_1] \rightsquigarrow Q'[p \mapsto O_1 \left[\overline{L}/x\right]]$, hence since the former term is s.n., so must be the latter, and hence also $O_1 \left[\overline{L}/x\right] \in \mathcal{SN}$; similarly, $O_2 \left[\overline{L}/x\right]$; then we can apply the IH with $(Q', p, O_1 \left[\overline{L}/x\right], O_2 \left[\overline{L}/x\right])$ to obtain the thesis.
- $Q'[p \mapsto O_1 \cup O_2]^\sigma$ (where $Q \overset{\sigma}{\rightsquigarrow} Q'$): by Lemma 17, we need to prove that, for all $q$ s.t. $\sigma(q) = p$, $Q'[q \mapsto O_1 \cup O_2] \in \mathcal{SN}$; since $Q[p \mapsto O_1] \in \mathcal{SN}$, we also have $Q'[p \mapsto O_1]^\sigma \in \mathcal{SN}$, which implies $Q'[q \mapsto O_1] \in \mathcal{SN}$ by Lemma 17; for the same reason, $Q'[q \mapsto O_2] \in \mathcal{SN}$; by Lemma 10, $\nu(Q') < \nu(Q)$, thus the thesis follows by IH.
- $Q_1[p \mapsto (\bigcup \{Q_2 | x \leftarrow O_1\}) \cup (\bigcup \{Q_2 | x \leftarrow O_2\})]$ (where $Q = Q_1 \circledast_p \bigcup \{Q_2 | x\}$): by Lemma 51, $\nu(Q_1) \leq \nu(Q)$; we also know $\|Q_1\|_p < \|Q\|_p$; take $O_1' := \bigcup \{Q_2 | x \leftarrow O_1\}$ and note that, since $Q[p \mapsto O_1] = Q_0[p \mapsto O_1']$, we have $O_1'$ is a subterm of a strongly normalizing term, thus $O_1' \in \mathcal{SN}$; similarly, we define $O_2' := \bigcup \{Q_2 | x \leftarrow O_2\}$ and show it is s.n. in a similar way; then $(Q_1, p, O_1', O_2')$ reduce the metric, and we can prove the thesis by IH.
- $Q_1[p \mapsto (\bigcup \{O_1 | x \leftarrow Q_2\}) \cup (\bigcup \{O_2 | x \leftarrow Q_2\})]$ (where $Q = Q_1 \circledast_p \bigcup \{x \leftarrow Q_2\}$): by Lemma 51, $\nu(Q_1) \leq \nu(Q)$; we also know $\|Q_1\|_p < \|Q\|_p$; take $O_1' := \bigcup \{O_1 | x \leftarrow Q_2\}$ and note that, since $Q[p \mapsto O_1] = Q_1[p \mapsto O_1']$, we have $O_1'$ is a subterm of a strongly normalizing term, thus $O_1' \in \mathcal{SN}$; similarly, we define $O_2' := \bigcup \{O_2 | x \leftarrow Q_2\}$ and show it is s.n. in a similar way; then $(Q_1, p, O_1', O_2')$ reduce the metric, and we can prove the thesis by IH.
- $Q_0[p \mapsto (\texttt{where}\ \overline{B}\ \texttt{do}\ O_1) \cup (\texttt{where}\ \overline{B}\ \texttt{do}\ O_2)]$ (where $Q = Q_0 \circledast_p \texttt{where}\ \overline{B}$): by Lemma 51, $\nu(Q_0) \leq \nu(Q)$; we also know $\|Q_0\|_p < \|Q\|_p$; take $O_1' := \texttt{where}\ B\ \texttt{do}\ O_1$ and note that, since $Q[p \mapsto O_1] = Q_0[p \mapsto O_1']$, we have $O_1'$ is a subterm of a strongly normalizing term, thus $O_1' \in \mathcal{SN}$; similarly, we define $O_2' := \texttt{where}\ \overline{B}\ \texttt{do}\ O_2$ and prove it is strongly normalizing in the same way; then $(Q_0, p, O_1', O_2')$ reduce the metric, and we can prove the thesis by IH.
- Contractions within $O_1$ or $O_2$ reduce $\nu(O_1) + \nu(O_2)$, thus the thesis follows by IH. ◀

Reducibility for conditionals similarly to comprehensions. However, to consider all the conversions commuting with `where`, we need to use the more general auxiliary continuations.

▶ **Lemma 57.** *If $Q[p \mapsto M \cup N] \in \mathcal{SN}$, then $Q[p \mapsto M] \in \mathcal{SN}$ and $Q[p \mapsto N] \in \mathcal{SN}$; furthermore, we have:*

$$\nu(Q[p \mapsto M]) \leq \nu(Q[p \mapsto M \cup N])$$
$$\nu(Q[p \mapsto N]) \leq \nu(Q[p \mapsto M \cup N])$$

**Proof.** We assume $p \in \mathsf{supp}(Q)$ (otherwise, $Q[p \mapsto M] = Q[p \mapsto N] = Q[p \mapsto M \cup N]$, and the thesis holds trivially), then we show that any contraction in $Q[p \mapsto M]$ has a corresponding non-empty reduction sequence in $Q[p \mapsto M \cup N]$, and the two reductions preserve the term form, therefore no reduction sequence of $Q[p \mapsto M]$ is longer than the maximal one in $Q[p \mapsto M \cup N]$. The same reasoning applies to $Q[p \mapsto N]$. ◀

**Proof of Lemma 34.** *Let $Q$, $B$, $O$ such that $Q[p \mapsto O] \in \mathcal{SN}$, $B \in \mathcal{SN}$, and $\mathsf{supp}(Q) \cap \mathsf{supp}(O) = \emptyset$. Then $Q[p \mapsto \texttt{where}\ B\ \texttt{do}\ O] \in \mathcal{SN}$.*

In this proof, we assume the names of bound variables are chosen so as to avoid duplicates, and distinct from the free variables. We proceed by well-founded induction on $(Q, B, O, p)$ using the following metric:

$$(Q_1, B_1, O_1, p_1) \prec (Q_2, B_2, O_2, p_2) \iff \\ (\nu(Q_1[p_1 \mapsto O_1]) + \nu(B_1), |Q_1|_{p_1}, \mathrm{size}(O_1)) \lessdot (\nu(Q_2[p_2 \mapsto O_2]) + \nu(B_2), |Q_2|_{p_2}, \mathrm{size}(O_2))$$

We will consider all possible contracta and show that each of them must be a strongly normalizing term; we will apply the induction hypothesis to new auxiliary continuations obtained by placing pieces of $O$ into $Q$ or vice-versa: the hypothesis on the supports of $Q$ and $O$ being disjoint is used to make sure that the new continuations do not contain duplicate holes and are thus well-formed. By cases on the possible contracta:

- $Q_1[q \mapsto Q_2 \left[\overline{L}/x\right]][p \mapsto (\texttt{where}\ B\ \texttt{do}\ O) \left[\overline{L}/x\right]]$, where $Q = (Q_1 \circledcirc_q \bigcup \{x \leftarrow \{\overline{L}\}\})[q \mapsto Q_2]$, $q \in \mathsf{supp}(Q_1)$, and $p \in \mathsf{supp}(Q_2)$; by the freshness condition we know $x \notin \mathrm{FV}(B)$, thus $(\texttt{where}\ B\ \texttt{do}\ O) \left[\overline{L}/x\right] = \texttt{where}\ B\ \texttt{do}\ (O \left[\overline{L}/x\right])$; we take $Q' = Q_1[q \mapsto Q_2 \left[\overline{L}/x\right]]$ and $O' = O \left[\overline{L}/x\right]$, and note that $\nu(Q'[p \mapsto O']) < \nu(Q[p \mapsto O])$, because the former term is a contractum of the latter: then we can apply the IH to prove $Q'[p \mapsto \texttt{where}\ B\ \texttt{do}\ O'] \in \mathcal{SN}$, as needed.
- $Q'[p \mapsto \texttt{where}\ B\ \texttt{do}\ O]^\sigma$, where $Q \overset{\sigma}{\rightsquigarrow} Q'$. We know $\nu(Q'[p \mapsto O]^\sigma) < \nu(Q[p \mapsto O])$ by Lemma 10 since the latter is a contractum of the former. By Lemma 17, for all $q$ s.t. $\sigma(q) = p$ we have $\nu(Q'[q \mapsto O]) \leq \nu(Q'[p \mapsto O]^\sigma)$; we can thus apply the IH to obtain $Q[q \mapsto \texttt{where}\ B\ \texttt{do}\ O] \in \mathcal{SN}$ whenever $\sigma(q) = p$. By Lemma 17, this implies the thesis.
- $Q_1[p \mapsto \texttt{where}\ B\ \texttt{do}\ \bigcup\{Q_2 | x \leftarrow O\}]$, where $Q = Q_1 \circledcirc_p \bigcup\{Q_2 | x\}$; we take $O' = \bigcup\{Q_2 | x \leftarrow O\}$, and we note that $Q[p \mapsto O] = Q_1[p \mapsto O']$ and $|Q_1|_p < |Q|_p$; we can thus apply the IH to prove $Q_1[p \mapsto \texttt{where}\ B\ \texttt{do}\ O'] \in \mathcal{SN}$, as needed.
- $Q[p \mapsto \emptyset]$, where $O = \emptyset$: this term is strongly normalizing by hypothesis.
- $Q[p \mapsto (\texttt{where}\ B\ \texttt{do}\ O_1) \cup (\texttt{where}\ B\ \texttt{do}\ O_2)]$, where $O = O_1 \cup O_2$; for $i = 1, 2$, we prove $Q[p \mapsto O_i] \in \mathcal{SN}$ and $\nu(Q[p \mapsto O_i]) \leq \nu(Q[p \mapsto O])$ by Lemma 30, and we also note $\mathrm{size}(O_i) < \mathrm{size}(O)$; then we can apply the IH to prove $Q[p \mapsto \texttt{where}\ B\ \texttt{do}\ O_i] \in \mathcal{SN}$, which implies the thesis by Lemma 30.
- $Q[p \mapsto \bigcup\{\texttt{where}\ B\ \texttt{do}\ O_1 | x \leftarrow O_2\}]$, where $O = \bigcup\{O_1 | x \leftarrow O_2\}$; we take $Q' = Q \circledcirc_p \bigcup\{x \leftarrow O_2\}$ and we have $Q'[p \mapsto \texttt{where}\ B\ \texttt{do}\ O_1] = Q[p \mapsto \bigcup\{\texttt{where}\ B\ \texttt{do}\ O_1 | x \leftarrow O_2\}]$; we thus note $\nu(Q'[p \mapsto O_1]) = \nu(Q[p \mapsto \bigcup\{O_1 | x \leftarrow O_2\}]) = \nu(Q[p \mapsto O])$, $|Q'|_p = |Q|_p$, and $\mathrm{size}(O_1) < \mathrm{size}(O)$, thus we can apply the IH to prove $Q'[p \mapsto \texttt{where}\ B\ \texttt{do}\ O_1] \in \mathcal{SN}$, as needed.
- reductions within $B$ or $O$ make the induction metric smaller, thus follow immediately from the IH. ◀