The Calculus of Linear Constructions — Technical Report

Qiancheng Fu

December 12, 2021

Contents

1 Introduction

This extended report is meant to accompany our paper of the same title. Here, we describe the meta-theory of CILC and their proofs in detail. All the results presented here have been formalized and proven correct in the Coq Proof Assistant.

2 Syntax of CLC (`clc_ast.v`)

i	$::=\ 0 \mid 1 \mid 2 \ldots$	universe levels
s, t	$::=\ U \mid L$	sorts
m, n, A, B, M	$::=\ U_i \mid L_i \mid x$	expressions
	$\mid\ (x :_s A) \to B$	
	$\mid\ (x :_s A) \multimap B$	
	$\mid\ \lambda x :_s A.n$	
	$\mid\ m\ n$	

3 Reduction and Equality of CLC (`clc_ast.v`)

$$\frac{m_1 \rightsquigarrow^* n \qquad m_2 \rightsquigarrow^* n}{m_1 \equiv m_2 : A}\textsc{Join} \qquad \frac{}{(\lambda x :_s A.m)\ n \rightsquigarrow m[n/x]}\textsc{Step-}\beta \qquad \frac{A \rightsquigarrow A'}{\lambda x :_s A.m \rightsquigarrow \lambda x :_s A'.m}\textsc{Step-}\lambda\textsc{L}$$

$$\frac{m \rightsquigarrow m'}{\lambda x :_s A.m \rightsquigarrow \lambda x :_s A.m'}\textsc{Step-}\lambda\textsc{R} \qquad \frac{A \rightsquigarrow_p A'}{(x :_s A) \to B \rightsquigarrow (x :_s A') \to B}\textsc{Step-L}\to$$

$$\frac{B \rightsquigarrow_p B'}{(x :_s A) \to B \rightsquigarrow (x :_s A) \to B'}\textsc{Step-R}\to \qquad \frac{A \rightsquigarrow_p A'}{(x :_s A) \multimap B \rightsquigarrow (x :_s A') \multimap B}\textsc{Step-L}\multimap$$

$$\frac{B \rightsquigarrow_p B'}{(x :_s A) \multimap B \rightsquigarrow (x :_s A) \multimap B'}\textsc{Step-R}\multimap \qquad \frac{m \rightsquigarrow m'}{m\ n \rightsquigarrow m'\ n}\textsc{Step-AppL} \qquad \frac{n \rightsquigarrow n'}{m\ n \rightsquigarrow m\ n'}\textsc{Step-AppR}$$

4 Confluence of CLC (`clc_confluence.v`)

4.1 Parallel Reduction

To prove the confluence property of CLC, we employ the standard technique utilizing parallel reductions.

$$\frac{}{x \rightsquigarrow_p x}\textsc{PStep-Var} \qquad \frac{}{s_i \rightsquigarrow_p s_i}\textsc{PStep-Sort} \qquad \frac{A \rightsquigarrow_p A' \qquad m \rightsquigarrow_p m'}{\lambda x :_s A.m \rightsquigarrow_p \lambda x :_s A'.m'}\textsc{PStep-}\lambda$$

$$\frac{m \rightsquigarrow_p m' \qquad n \rightsquigarrow_p n'}{m\ n \rightsquigarrow_p m'\ n'}\textsc{PStep-App} \qquad \frac{m \rightsquigarrow_p m' \qquad n \rightsquigarrow_p n'}{(\lambda x :_s A.m)\ n \rightsquigarrow_p m'[n'/x]}\textsc{PStep-}\beta$$

$$\frac{A \rightsquigarrow_p A' \qquad B \rightsquigarrow_p B'}{(x :_s A) \to B \rightsquigarrow_p (x :_s A') \to B'}\textsc{PStep}\to \qquad \frac{A \rightsquigarrow_p A' \qquad B \rightsquigarrow_p B'}{(x :_s A) \multimap B \rightsquigarrow_p (x :_s A') \multimap B'}\textsc{PStep}\multimap$$

4.2 Reduction Lemmas

Here, we prove some simple lemmas concerning $\rightsquigarrow$, $\rightsquigarrow^*$ and substitution.

Definition 4.1. For a term m and a map σ from variables to terms, let $m[\sigma]$ be the term obtained by applying σ uniformly to all free variables in m.

Definition 4.2. For maps σ, τ from variables to terms, we say that σ reduces to τ if for any variable x there exists a reduction $(\sigma\ x) \rightsquigarrow^* (\tau\ x)$. We write $\sigma \rightsquigarrow^* \tau$ when it is clear from context that σ, τ are maps and not terms.

Lemma 4.1. *For terms m, n and a map σ from variables to terms, if there exist a step $m \rightsquigarrow n$, then there exists a step $m[\sigma] \rightsquigarrow n[\sigma]$.*

Proof. By induction on the derivation of $m \rightsquigarrow n$. ☐

Lemma 4.2. *For terms m_1, m_2, n_1, n_2, if these exists reductions $m_1 \rightsquigarrow^* m_2$ and $n_1 \rightsquigarrow^* n_2$, then there exists reduction $(m_1\ n_1) \rightsquigarrow^* (m_2\ n_2)$.*

Proof. By transitivity of $\rightsquigarrow^*$ and applying rules Step-AppL, Step-AppR. ☐

Lemma 4.3. *For terms A_1, A_2, m_1, m_2 and sort s, if there exists reductions $A_1 \rightsquigarrow^* A_2$ and $m_1 \rightsquigarrow^* m_2$, then there exists reduction $\lambda x :_s A_1.m_1 \rightsquigarrow^* \lambda x :_s A_2.m_2$.*

Proof. By transitivity of $\rightsquigarrow^*$ and applying rules Step-λL, Step-λR. ☐

Lemma 4.4. *For terms A_1, A_2, B_1, B_2 and sort s, if there exists reductions $A_1 \rightsquigarrow^* A_2$ and $B_1 \rightsquigarrow^* B_2$, then there exists reduction $(x :_s A_1) \to B_1 \rightsquigarrow^* (x :_s A_2) \to B_2$.*

Proof. By transitivity of $\rightsquigarrow^*$ and applying rules Step-L→, Step-R→. ☐

Lemma 4.5. *For terms A_1, A_2, B_1, B_2 and sort s, if there exists reductions $A_1 \rightsquigarrow^* A_2$ and $B_1 \rightsquigarrow^* B_2$, then there exists reduction $(x :_s A_1) \multimap B_1 \rightsquigarrow^* (x :_s A_2) \multimap B_2$.*

Proof. By transitivity of $\rightsquigarrow^*$ and applying rules Step-L⊸, Step-R⊸. ☐

Lemma 4.6. *For terms m, n and a map σ from variables to terms, if there exist a reduction $m \rightsquigarrow^* n$, then there exists a reduction $m[\sigma] \rightsquigarrow^* n[\sigma]$.*

Proof. By induction on the derivation of $\rightsquigarrow^*$, the transitivity of $\rightsquigarrow^*$ and Lemma 4.1. ☐

Lemma 4.7. *For maps σ, τ from variables to terms, if there is a map reduction $\sigma \rightsquigarrow^* \tau$, then for any term m these is a reduction $m[\sigma] \rightsquigarrow^* m[\tau]$.*

Proof. By induction on the structure of m, applying Lemmas 4.2, 4.3, 4.4, 4.5. ☐

4.3 Equality Lemmas

Here, we prove some simple lemmas concerning $\rightsquigarrow^*$, $\equiv$ and substitution.

Definition 4.3. For maps σ, τ from variables to terms, we say that σ is equal to τ if for any variable x there exists an equality $(\sigma\ x) \equiv (\tau\ x)$. We write $\sigma \equiv \tau$ when it is clear from context that σ, τ are maps and not terms.

Lemma 4.8. *For any map f from terms to terms, if for any terms m, n such that $m \rightsquigarrow n$ implies $f\ m \equiv f\ n$, then for any terms m, n equality $m \equiv n$ implies $f\ m \equiv f\ n$.*

Proof. By the properties of the transitive reflexive closure $\rightsquigarrow^*$ and that $\equiv$ is an equivalence relation. ☐

Lemma 4.9. *For terms m_1, m_2, n_1, n_2, if there exists equalities $m_1 \equiv m_2$ and $n_1 \equiv n_2$, then there exists equality $(m_1\ n_1) \equiv (m_2\ n_2)$.*

Proof. By transitivity of $\equiv$ and applying rules Join, Step-AppL, Step-AppR. ☐

Lemma 4.10. *For terms A_1, A_2, m_1, m_2 and sort s, if there exists equalities $A_1 \equiv A_2$ and $m_1 \equiv m_2$, then there exists equality $\lambda x :_s A_1.m_1 \equiv \lambda x :_s A_2.m_2$.*

Proof. By transitivity of $\equiv$ and applying rules Join, Step-λL, Step-λR. ☐

Lemma 4.11. *For terms A_1, A_2, B_1, B_2 and sort s, if there exists equalities $A_1 \equiv A_2$ and $B_1 \equiv B_2$, then there exists equality $(x :_s A_1) \to B_1 \equiv (x :_s A_2) \to B_2$.*

Proof. By transitivity of $\equiv$ and applying rules Join, Step-L→, Step-R→. ☐

Lemma 4.12. *For terms A_1, A_2, B_1, B_2 and sort s, if there exists equalities $A_1 \equiv A_2$ and $B_1 \equiv B_2$, then there exists equality $(x :_s A_1) \multimap B_1 \equiv (x :_s A_2) \multimap B_2$.*

Proof. By transitivity of $\equiv$ and applying rules Join, Step-L⊸, Step-R⊸. ☐

Lemma 4.13. *For terms m, n and map σ from variables to terms, if there is equality $m \equiv n$, then there is equality $m[\sigma] \equiv n[\sigma]$.*

Proof. By Lemmas 4.8 and 4.1. ☐

Lemma 4.14. *For maps σ, τ from variables to terms and term m, if these is map equality $\sigma \equiv \tau$, then there is equality $m[\sigma] \equiv m[\tau]$.*

Proof. By induction on the structure of m, applying Lemmas 4.9, 4.10, 4.11, 4.12. □

Lemma 4.15. *For terms m_1, m_2, n, if there is equality $m_1 \equiv m_2$, then there is equality $n[m_1/x] \equiv n[m_2/x]$ for any variable $x \in FV(n)$.*

Proof. This is a special case of Lemma 4.14 where σ maps x to m_1 and τ maps x to m_2. □

4.4 Parallel Reduction Lemmas

Definition 4.4. For maps σ, τ from variables to terms, we say σ parallel reduces to τ if for any variable x there exists a parallel reduction $(\sigma\ x) \leadsto_p (\tau\ x)$. We write $\sigma \leadsto_p \tau$ when it is clear from context that σ, τ are maps and not terms.

Lemma 4.16. *For any term m, there exists a reflexive parallel reduction $m \leadsto_p m$.*

Proof. By induction on the structure of m. □

Lemma 4.17. *For any map σ from variables to terms, there exists a reflexive parallel map reduction $\sigma \leadsto_p \sigma$.*

Proof. By Definition 4.4 and Lemma 4.16. □

Lemma 4.18. *For any terms m, n, if there exists step $m \leadsto n$, then there exists a parallel reduction $m \leadsto_p n$.*

Proof. By induction on the derivation of $m \leadsto n$ and Lemma 4.16. □

Lemma 4.19. *For terms m, n, if there exists parallel reduction $m \leadsto_p n$, then there exists a reduction $m \leadsto^* n$.*

Proof. By induction on the derivation of $m \leadsto_p n$, utilizing the transitive property of $\leadsto^*$ and Lemmas 4.2, 4.3, 4.4, 4.5, 4.6, 4.7. □

Lemma 4.20. *For terms m, n and map σ from variables to terms, if there exists parallel reduction $m \leadsto_p n$, there exists parallel reduction $m[\sigma] \leadsto_p n[\sigma]$.*

Proof. By induction on the derivation of $m \leadsto_p n$ and Lemma 4.16. □

Lemma 4.21. *For terms m, n and maps σ, τ from variables to terms, if there exists parallel reduction $m \leadsto_p n$ and parallel map reduction $\sigma \leadsto_p \tau$, there exists parallel reduction $m[\sigma] \leadsto_p n[\tau]$.*

Proof. By induction on the derivation of $m \leadsto_p n$. □

Lemma 4.22. *For terms m_1, m_2, n, if there is parallel reduction $m_1 \leadsto_p m_2$, then there is parallel reduction $n[m_1/x] \leadsto_p n[m_2/x]$ for any variable $x \in FV(n)$.*

Proof. By Lemma 4.16, this is a special case of Lemma 4.21 where σ maps x to m_1 and τ maps x to m_2. □

4.5 Confluence Theorem

We first show that $\leadsto_p$ satisfies the diamond property. Using the diamond property, we ultimately prove the confluence theorem.

Lemma 4.23. *CLC term reduction has the diamond property. For terms m, m_1, m_2, if there are parallel reductions $m \leadsto_p m_1$ and $m \leadsto_p m_2$, then there exists term m' such that $m_1 \leadsto_p m'$ and $m_2 \leadsto_p m'$.*

Proof. By induction on the derivation of $m \leadsto_p m_1$. Each case in the induction specializes m appearing in $m \leadsto_p m_2$, allowing one to invert its derivation in a syntax directed way and apply the induction hypothesis. The difficult cases are due to PStep-β as it concerns substitution, so Lemma 4.21 is used to push these cases through. □

Lemma 4.24. *Strip lemma. For terms m, m_1, m_2, if there is parallel reduction $m \rightsquigarrow_p m_1$ and reduction $m \rightsquigarrow^* m_2$, then there exists term m' such that $m_1 \rightsquigarrow^* m'$ and $m_2 \rightsquigarrow_p m'$.*

Proof. By induction on the derivation of $m \rightsquigarrow_p m_1$, utilizing transitivity of $\rightsquigarrow^*$ and Lemmas 4.18, 4.19, 4.23. ☐

Theorem 4.25. *CLC term reduction is confluent. For terms m, m_1, m_2, if there are reductions $m \rightsquigarrow^* m_1$ and $m \rightsquigarrow^* m_2$, then there exists term m' such that $m_1 \rightsquigarrow^* m'$ and $m_2 \rightsquigarrow^* m'$.*

Proof. By induction on the derivation of $m \rightsquigarrow^* m_1$, utilizing transitivity of $\rightsquigarrow^*$ and Lemmas 4.18, 4.19, 4.24. ☐

4.6 Corollaries of Confluence

The following results are all corollaries of confluence, proven using a combination of induction, transitivity and confluence. These corollaries allow us to refute false reductions and equalities in future proofs.

Corollary 4.25.1. *For a universe s_i and term m, if there is reduction $s_i \rightsquigarrow^* m$, then $m = s_i$.*

Corollary 4.25.2. *For variable x and term m, if there is reduction $x \rightsquigarrow^* m$, then $m = x$.*

Corollary 4.25.3. *For terms A, B, m and sort s, if there is reduction $(x :_s A) \to B \rightsquigarrow^* m$, then there exists A', B' such that there are reductions $A \rightsquigarrow^* A'$, $B \rightsquigarrow^* B'$ and $m = (x :_s A') \to B'$.*

Corollary 4.25.4. *For terms A, B, m and sort s, if there is reduction $(x :_s A) \multimap B \rightsquigarrow^* m$, then there exists A', B' such that there are reductions $A \rightsquigarrow^* A'$, $B \rightsquigarrow^* B'$ and $m = (x :_s A') \multimap B'$.*

Corollary 4.25.5. *For terms A, m, n and sort s, if there is reduction $\lambda x :_s A.m \rightsquigarrow^* n$, then there exists A', m' such that there are reductions $A \rightsquigarrow^* A'$, $m \rightsquigarrow^* m'$ and $n = \lambda x :_s A'.m'$.*

Corollary 4.25.6. *For sorts s, t and levels i, j, if there is equality $s_i \equiv t_j$, then there is $s = t$ and $i = j$.*

Corollary 4.25.7. *For terms A_1, A_2, B_1, B_2 and sorts s, t, if there is equality $(x :_s A_1) \to B_1 \equiv (x :_t A_2) \to B_2$, then there are equalities $A_1 \equiv A_2$, $B_1 \equiv B_2$ and $s = t$.*

Corollary 4.25.8. *For terms A_1, A_2, B_1, B_2 and sorts s, t, if there is equality $(x :_s A_1) \multimap B_1 \equiv (x :_t A_2) \multimap B_2$, then there are equalities $A_1 \equiv A_2$, $B_1 \equiv B_2$ and $s = t$.*

5 Context of CLC (`clc_context.v`)

Contexts of CLC are of the form $x_1 :_{s_1} A_1, x_2 :_{s_2} A_2, ... x_k :_{s_k} A_k$ where each free variable x_i is assigned a type A_i and sort s_i. Contexts will be referred to by meta variables Γ and Δ.

$$\frac{}{\epsilon \vdash}\textsc{Wf-}\epsilon \qquad \frac{\Gamma \vdash \qquad \overline{\Gamma} \vdash A : U_i}{\Gamma, x :_U A \vdash}\textsc{Wf-U} \qquad \frac{\Gamma \vdash \qquad \overline{\Gamma} \vdash A : L_i}{\Gamma, x :_L A \vdash}\textsc{Wf-L}$$

$$\frac{}{|\epsilon|}\textsc{Pure-}\epsilon \qquad \frac{|\Gamma| \qquad \Gamma \vdash A : U_i}{|\Gamma, x :_U A|}\textsc{Pure-U}$$

$$\frac{}{\epsilon \ddagger \epsilon \ddagger \epsilon}\textsc{Merge-}\epsilon \qquad \frac{\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma}{\Gamma_1, x :_U A \ddagger \Gamma_2, x :_U A \ddagger \Gamma, x :_U A}\textsc{Merge-U}$$

$$\frac{\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma \qquad x \notin \Gamma_2}{\Gamma_1, x :_L A \ddagger \Gamma_2 \ddagger \Gamma, x :_L A}\textsc{Merge-L1} \qquad \frac{\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma \qquad x \notin \Gamma_1}{\Gamma_1 \ddagger \Gamma_2, x :_L A \ddagger \Gamma, x :_L A}\textsc{Merge-L2}$$

5.1 Merge Lemmas

Since weakening and contraction rules will not be allowed on restricted variables, it is necessary to have lemmas that enable the manipulation of contexts.

Lemma 5.1. *For contexts* $\Gamma_1, \Gamma_2, \Gamma$, *if there is* $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$, *then there is* $\Gamma_2 \ddagger \Gamma_1 \ddagger \Gamma$.

Proof. By induction on the derivation of $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$. □

Lemma 5.2. *For any context* Γ, *if there is* $|\Gamma|$, *then there is* $\Gamma \ddagger \Gamma \ddagger \Gamma$.

Proof. By induction on the derivation of $|\Gamma|$. □

Lemma 5.3. *For any context* Γ, *there is* $\overline{\Gamma} \ddagger \Gamma \ddagger \Gamma$.

Proof. By induction on the structure of Γ. □

Lemma 5.4. *For any context* Γ, *there is* $\Gamma \ddagger \overline{\Gamma} \ddagger \Gamma$.

Proof. By induction on the structure of Γ. □

Lemma 5.5. *For contexts* $\Gamma_1, \Gamma_2, \Gamma$, *if there is* $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$ *and* $|\Gamma|$, *then there is* $|\Gamma_1|$ *and* $|\Gamma_2|$.

Proof. By induction on the derivation of $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$. □

Lemma 5.6. *For contexts* $\Gamma_1, \Gamma_2, \Gamma$, *if there is* $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$ *and* $|\Gamma_1|$, *then there is* $\Gamma = \Gamma_2$.

Proof. By induction on the derivation of $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$. □

Lemma 5.7. *For contexts* $\Gamma_1, \Gamma_2, \Gamma$, *if there is* $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$ *and* $|\Gamma_2|$, *then there is* $\Gamma = \Gamma_1$.

Proof. By induction on the derivation of $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$. □

Lemma 5.8. *For contexts* $\Gamma_1, \Gamma_2, \Gamma$, *if there is* $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$, *and also* $|\Gamma_1|$, $|\Gamma_2|$, *then there is* $|\Gamma|$.

Proof. By induction on the derivation of $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$. □

Lemma 5.9. *For contexts* $\Gamma_1, \Gamma_2, \Gamma$, *if there is* $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$, *and also* $|\Gamma_1|$, $|\Gamma_2|$, *then there is* $\Gamma_1 = \Gamma_2$.

Proof. By induction on the derivation of $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$. □

Lemma 5.10. *For contexts* $\Gamma_1, \Gamma_2, \Gamma$, *if there is* $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$, *then there is* $\overline{\Gamma_1} = \overline{\Gamma}$ *and* $\overline{\Gamma_2} = \overline{\Gamma}$.

Proof. By induction on the derivation of $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$. □

Lemma 5.11. *For any context* Γ, *there is* $\overline{\Gamma} \ddagger \overline{\Gamma} \ddagger \overline{\Gamma}$.

Proof. By induction on the structure of Γ. □

Lemma 5.12. *For contexts* $\Gamma_1, \Gamma_2, \Gamma, \Delta_1, \Delta_2$, *if there is* $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$ *and* $\Delta_1 \ddagger \Delta_2 \ddagger \Gamma_1$, *then there exists* Δ *such that* $\Delta_1 \ddagger \Gamma_2 \ddagger \Delta$ *and* $\Delta \ddagger \Delta_2 \ddagger \Gamma$.

Proof. By induction on the derivation of $\Gamma_1 \ddagger \Gamma_2 \ddagger \Gamma$. □

5.2 Restriction and Purity Lemmas

Lemma 5.13. *For any context* Γ, *there is* $\overline{\Gamma} = \overline{\overline{\Gamma}}$.

Proof. By induction on the structure of Γ. □

Lemma 5.14. *For any context* Γ, *if there is* $|\Gamma|$, *then there is* $\Gamma = \overline{\Gamma}$.

Proof. By induction on the structure of Γ. □

Lemma 5.15. *For any context* Γ , *there is* $|\overline{\Gamma}|$.

Proof. By induction on the structure of Γ. □