An Interactive Proof Assistant for Linear Logic

by

Maksim Trifunovski

A thesis submitted to the
faculty of Wesleyan University
in partial fulfillment of the requirements for the
Degree of Bachelor of Arts
with Departmental Honors in Mathematics and Computer Science

"When you want something, all the universe conspires in helping you to achieve it."

Paulo Coelho, The Alchemist

Acknowledgements

First of all I would like to thank my research advisor Dan Licata. Dan took me under his wing when I was just a freshman who thought Computer Science is all about implementing algorithms in C++. Over the years Dan has taught me how to teach, how to have patience and how to think. He taught me everything from the basics of functional programming, up to category theory. Thank you for being patient even when I was not grasping the simplest of things, and thank you for your excitement when I made even the slightest progress on my research.

I would also like to thank Prof. Norman Danner and Prof. Edward Morehouse for taking the time to read and evaluate this thesis. Thank you to Prof. Jim Lipton for giving me the first flavors of designing programming languages and Prof. Philip Scowcroft for introducing me to logic.

Thank you to Joomy Korkut for being the best research partner one could ask for. Exploring Agda with you was one of the reasons why I decided to write this thesis, and I could not have finished it without your advice on all of the issues that came up during the implementation.

Thank you to Angus, Max, Varun, Ivona, Maxine, Noelle, Yulia, Pi, Emily and all of my other friends who supported me in writing this thesis and kept me thinking positively even in the direst of moments.

Thank you to my father who taught me to believe in myself and to never give up no matter how daunting a challenge looks. Thank you to my sister for being my biggest supporter and helping me get through the toughest periods of my life.

Thank you to my aunt for always being there for me, even when facing troubles I cannot share with anyone else.

Lastly, thank you to my mother who made me fall in love with math during my early ages, and made sure that the love will last forever. This thesis and proof assistant are dedicated to her.

Abstract

Building formal proofs is made easier by tools called proof assistants. In this thesis we present the process of building a proof assistant for propositional intuitionistic linear logic. Linear logic is a refinement of classical and intuitionistic logic which puts its main focus on the role of its formulas as resources. While the consumption of formulas as if they are resources is a big advantage of linear logic that other logics do not offer, it is also a burden when trying to build proofs. Resource allocation in rules like Tensor Right is a difficult task since we need to predict where resources are going to be used by the rest of the proof before building it. Previous work by Hodas and Miller presents a way of using Input-Output contexts to work around this issue. But because we are building a proof-assistant and not an automated theorem prover, we need to allow the user to be able to switch between goals at any time, and be able to construct the proof in any order they want, so we cannot solve the problem with such contexts. We tackle this problem by allowing unbounded context growth when moving up the proof derivation tree, and instead allowing terms to only use variables from a given resources multiset which is a subset of the whole context. These subsets then have to satisfy a given set of equations that make them suitable for simulating context splittings and changes. In order to allow for incremental proof construction we use meta variables to stand for incomplete terms and modal contexts to store said variables, which builds on previous work by Nanevski et al. We define two sequent calculi, a base one, and one that represents the implementation. We present a cut admissibility proof that

proves that our base sequent calculus is consistent, as well as a theorem which shows that every proof in the implementation calculus, has a proof in the base calculus as well.

Contents

1. Introduction

1.1. Background.

1.1.1. *Proof assistants.*

A proof assistant, also called an interactive theorem prover is a piece of software made to assist a human user with the development of formal proofs. These tools usually come with some sort of interactive proof editor, with which the user can guide the search for proofs. The computer helps by providing details and sometimes steps on how to advance the search.

The history of proof assistants goes back to the early 70s when Milner presented LCF (Logic for Computable Functions), which is a proof-checking program proposed by Dana Scott in 1969. [7] The program was defined with the goal of allowing users to interactively generate formal proofs about computable functions and functionals, while helping them with a subgoaling facility and a powerful simplification mechanism. LCF opened the doors for many new proof assistants.

Coq is a proof assistant which supports dependent types, uses metavariables for proof construction, and was used to create a surveyable proof of the four color theorem.[3]. Agda, a dependently typed functional programming language and a proof assistant, was developed by Norell [9] and also heavily relies on metavariables. In Agda we can write functions like:

$$\text{add} : \mathbb{N} \to \mathbb{N} \to \mathbb{N}$$

$$\text{add x y} = ?$$

In this function definition, the '?' stands for a metavariable. When interacting with this metavariable, Agda will show the user the expected type and allow the user to refine the metavariable (replace it with another term). By using metavariables, a proof assistant allows for incremental program construction.

Proof assistants are becoming increasingly more popular in both the Mathematics and Computer Science communities, as they help users substantially when building proofs. The proof assistant can first give the user the general structure of a proof, and help the user see what the smaller components of the proof that need to be completed are. For each of these smaller components, the proof assistant will try to offer the user as much valuable information possible, like types and variables available, which further helps advance the proof construction. Probably the most useful characteristic of proof assistants is fast proof checking, something that would take many tedious rounds of human checks.

1.1.2. *Sequent Calculus.*

A formal system is a well-defined system which is made up of the following:

1. A finite set of symbols used in the construction of formulas.
2. A grammar which shows how the formulas are constructed out of the symbols.
3. A set of axioms.
4. A set of inference rules.

Sequent calculus is a family of formal systems which share a particular style of inference and properties. The first such systems were introduced by Gerhard Gentzen in 1934/1935.[1] With their introduction Gentzen also introduced his "Main Theorem", now known as the cut-elimination theorem, which has many consequences, one of which is consistency of the system.

In a Gentzen style sequent calculus, every line is a conditional theorem with zero or more conditions on the left and zero or more asserted propositions on the right. A sequent is an object of the form

$$A_1, ..., A_n \vdash B_1, ..., B_m$$

where the formulas on the left of $\vdash$ are called the antecedents, and the formulas on the right are called consequents. Each of A_i, B_j is a formula, and $n, m \geq 0$. We usually write Γ for the set of all antecedents.

An inference rule now is an object which contains of zero or more premises and one conclusion, where both the premises and the conclusion are sequents. A proof is a derivation tree where the nodes are inference rules and the leafs are axioms. We give as an example the proof of the sequent $A, B \vdash A \wedge B$ in intuitionistic logic.

$$\dfrac{\dfrac{}{A, B \vdash A}\, A \in A, B \qquad \dfrac{}{A, B \vdash B}\, B \in A, B}{A, B \vdash A \wedge B}\, \wedge_R$$

1.1.3. *Linear Logic.*

Linear logic is a substructural logic proposed by Jean-Yves Girard as a refinement of classical and intuitionistic logic.[2] Ideas from linear logic have been influential in fields such as programming languages, game semantics, quantum physics and linguistics primarily because of its emphasis on resource-boundedness, duality and interaction. Contrary to classical and intuitionistic logic, we no longer have an ever-expanding collection of persistent "truths", as we now care about manipulating resources that cannot be duplicated or thrown away at will.

In classical and intuitionistic logics we deal with stable truths:

if A and $A \Rightarrow B$, then B, but A still holds.

In linear logic such a proof would consume both A and $A \Rightarrow B$ in the process. Since linear logic is called the logic of resources, we have to introduce a new sort of implication, called linear implication ($\multimap$). Now if we had \$1 and a discount coupon that allows us to buy a lollipop for \$1, we could buy the lollipop, but in the process we will get rid of both the \$1 and the coupon, hence use up all of our resources. We can display the proof as follows:

$$\dfrac{\dfrac{}{\$1 \vdash \$1}\,\text{ax} \qquad \dfrac{}{L \vdash L}\,\text{ax}}{\$1, \$1 \multimap L \vdash L}\,\multimap_R$$

The proof uses the $\multimap_R$ inference rule which says that if our resources are $\Gamma, \Delta, A \multimap B$, and from Γ we can prove A while from Δ, B we can prove C, then we can prove C from $\Gamma, \Delta, A \multimap B$.

Linear logic also introduces two conjunctions, $\otimes$ (times) and & (with). Both of them express the availability of two actions, but in the case of $\otimes$ we have the resources to do both, while in the case of & we only have the resources to do one, and we get to choose which one. Consider the following example:

A : spend \$ 1,

B : buy a lollipop

C : buy chewing gum

An action of type A will be removing \$1 from our wallet. An action of type $A \multimap B$ means spending \$ 1 in order to get a lollipop. Given an action of type $A \multimap B$ and an action of type $A \multimap C$, there is no way to form an action of type $A \multimap (B \otimes C)$, since \$ 1 is not enough resources to get both a lollipop and a piece of gum. In order to get both we would need $A \otimes A$ as our resources, and the proof follows below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A} \quad \dfrac{}{B \vdash B}}{A \multimap B, A \vdash B} \multimap_L \quad \dfrac{\dfrac{}{A \vdash A} \quad \dfrac{}{B \vdash B}}{A \multimap C, A \vdash C} \multimap_L}{A \multimap B, A \multimap C, A, A \vdash B \otimes C} \otimes_R}{A \multimap B, A \multimap C, A \otimes A \vdash B \otimes C} \otimes_L}{A \multimap B, A \multimap C \vdash (A \otimes A) \multimap (B \otimes C)} \multimap_R$$

To express choice of one action between two, without giving us the freedom to choose, linear logic uses $\oplus$ (plus). In terms of computer science, the distinction $\&/\oplus$ corresponds to the distinction outer/inner non-determinism.

Linear logic has a second disjunction, "$\parr$" (par) which is the dual of "$\otimes$", and expresses a dependency between two types of actions. We can read $A \parr B$ as either $A^\perp \multimap B$ or as $B^\perp \multimap A$, i.e. "$\parr$" is a symmetric form of "$\multimap$".

The linear negation which we have just mentioned above, behaves like transposition in linear algebra, i.e. it expresses a duality:

$$\text{action of type } A = \text{reaction of type } A^\perp$$

The main property of linear negation is that $A^{\perp\perp}$ can be identified with A just like in classical logic.

Intuitionistic linear logic is the intuitionistic restriction of linear logic. The sequent calculus of ILL is obtained from the two-sided sequent calculus of linear logic by constraining sequents to have exactly one formula on the right-hand side $\Gamma \vdash A$. The connectives $\parr, \perp$ and ? are not available anymore, but linear implication $\multimap$ is. The two example proofs we displayed above were produced in this sequent calculus.

1.1.4. *Curry-Howard correspondence.*

The Curry-Howard correspondence, also called the Curry-Howard isomorphism, represents the direct relationship between computer programs and mathematical proofs. [6] Applied to a sequent calculus it maps the right introduction rules on the logic side to constructors of code on the programming side, left introduction rules to constructors of evaluation stacks and cut elimination to reduction in a form of abstract machine.

We apply the Curry-Howard correspondence to an enrichened subset of intuitionistic linear logic with metavariables, in order to get the sequent calculus in Figure 13.

1.2. Current work.

Inspired by the power of modern proof assistants like Agda and Coq, the research project which this thesis is based on was undertaken with a goal of formulating a framework that can be used to implement a proof assistant based on many different logics.

We chose to build a proof assistant for propositional intuitionstic linear logic, so we enrichened its sequent calculus with modal contexts and metavariables and then applied the Curry-Howard correspondence to it to get terms for each type, and present that calculus as our base logical calculus.

Dealing with linear logic introduces issues that do not appear when building proof assistants for other logics. One such issue is resource allocation during rules that split the context of variables. To solve this we decide to let contexts grow unbounded while moving up the derivation tree, while introducing resource multisets, which act as subsets of the contexts under which they operate, but also satisfy a given set of equations.

Using metavariables helps us construct a proof step by step, by acting as placeholders for terms we have not built yet. We prove that our system is consistent by providing a cut admissibility proof, and then we provide an implementation calculus which mirrors it.

In our implementation calculus we introduce resource variables which show up in the place of resource multisets. We have a resource restrictions context which contains the restrictions that each variable has to satisfy. Because we keep this context consistent, it can also identify resource multisets that should show up in the locations of the resources variables, when we are converting a proof from this calculus to the base one. We came up with this implementation because the main idea of a proof-assistant is to build the proof of a theorem in any order the user desires. Such a process is not feasable when using Input-Output contexts (which prove the left-most branch of the proof with a subset of all resources, and then provide the output resources to the rest of the branches of the proof), nor is it efficient to split linear contexts, since a lot of decisions on resource allocation would have to be made before hand. In our implementation we allow all goals to use all possible unused resources at any time, and as soon as a resource has been used in any of the linked goals, we remove it from the available resources for the other goals linked with our primary one. The reader can find the implementation at https://github.com/trifunovski/Janet.

We provide a theorem that we can convert any proof from the implementation calculus under a consistent restrictions context to a proof in the base calculus.

In the conclusion we present our future ideas for how this proof assistant can be improved, and how the framework can be used to build other proof assistants.

1.3. Related work.

Nanevski et al. present a Contextual Modal Type Theory [8]. They provide a sequent calculus where judgments have 2 types of contexts, a context of propositions and a modal context of metavariables. Then they prove admissibility of cut for both contexts. They continue to extend their type theory with dependent types and talk about an implementation of meta-variables in logical frameworks. The work in this thesis is based on the work by Nanevski et al.

Shack-Nielsen and Schürman present a Linear Contextual Modal Type Theory that builds on the work by Nanevski et al. [10] They provide a type theory for a subset of linear logic enrichened with modal contexts and metavariables. The subset they choose to work with differs from the one considered in this thesis, but it also tackles the issue of assigning specific resources to different parts of the proof. They use linear contexts which get split in order to deal with resource allocation. We instead let contexts grow unbounded while using resource multisets to deal with resource allocation. A contextual modal substitution (or refining a goal) is also presented.

In his PhD Thesis [4], Hodas presents the theory and design of a logic programming language based on linear logic. Hodas also uses Intuitionistic Linear Logic but works towards creating a goal-directed proof-search program, which tries to find a proof of a goal without the help of human interaction. Hodas encounters similar challenges to ours, with the main one being resource management. Hodas notes that these issues can cripple a naive implementation of this logic, hence he proposes an implementation that circumvents these problems by delaying some choices in the proof search.

This model of resource consumption was first displayed in an article by Hodas and Miller [5]. They present a so-called IO-context, in which $I - O \equiv \Gamma$ and if A is a goal formula then $I\{A\}O$ is provable if and only if $\Gamma \vdash A$ has a proof. This makes finding a proof for a tensor goal easy, since we use up as many of the resources as needed in proving the first type, and then try to find a proof of the second type using the rest of the resources, and saves us from making a decision on which resources go where upfront. Because we are building a proof-assistant and not an automated theorem prover, we need to allow the user to be able to switch between goals at any time, and be able to construct the proof in any order they want. In order to do that we use resource variables which at any point can identify an upper bound resources multiset. This multiset tells us what variables we can use in the current branch of our proof. Since the resource variables are linked using restriction equations, using up variables in one, removes those variables from the upper bound multiset of another.

This thesis drew a lot of inspiration from Agda, a proof assistant designed by Norell in his PhD Thesis [9]. The idea of using metavariables to stand for parts of the term that still haven't been built is being used heavily in Agda, as it allows for incremental proof construction.

2. Janet: An Interactive Theorem Prover

Because of the Curry-Howard isomorphism, the equivalent of a proof of a theorem in propositional linear logic is a construction of a term that has the type equivalent to the right-hand side of the theorem, using as resources variables which have types as the left-hand side of the theorem.

Therefore building a proof in Janet starts by the user entering their context of variables with their types. [Figure 1]

FIGURE 1. Program start

The next step is to enter the type which is equivalent to the right-hand side of the theorem. [Figure 2]

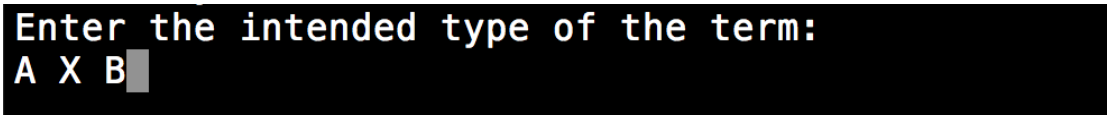

FIGURE 2. Specifying the type

Once the sequent has been set up, we go in a loop that follows the following steps:

1. Ask the user for a goal to work on. [Figure 3]

```
y : B , x : A  ⊢ { ?1 } : (A ⊗ B)
Enter the desired hole #:
1
```

FIGURE 3. Selecting the goal

2a. Display that goal's type and resources, as well as rules the user can use to refine it. [Figure 4]

```
y : B , x : A  ⊢ { ?1 } : (A ⊗ B)
Enter the desired hole #:
1
Hole 1 was selected, with type (A ⊗ B)
You can use the following variables: y@2 : B, x@1 : A
You can use the following meta-variables: 1@4
You can use the following rules: MV, Xright
Select a rule to be applied:
Xright
```

FIGURE 4. Selecting the rule to refine the goal

2b. If a left rule is used, select the variable to apply it to. [Figure 5]

3. Stop if we end up with a term that doesn't contain any new goals after applying the rule. Otherwise the old goal has been replaced by a term that contains new goals, so we loop back to 1 [Figure 6].

```
y : A , f : (A ⊸ B)  ⊢ { ?1 } : B
Enter the desired hole #:
1
Hole 1 was selected, with type B
You can use the following variables: y@2 : A, f@1 : (A ⊸ B)
You can use the following meta-variables: 1@4
You can use the following rules: MV, -oleft
Select a rule to be applied:
-oleft
Select the variable to which to apply the rule:
f@1
```

FIGURE 5. Selecting a variable to which the rule will be applied

---

```
y : A , f : (A ⊸ B)  ⊢ let x be (f) ({ ?2 }) in { ?3 } : B
Enter the desired hole #:
```

FIGURE 6. Looping back with new goals in the term

```
y : B , x : A  ⊢ (x , y) : (A ⊗ B)
We are done!
```

FIGURE 7. Successfully constructing a term

Once we end up with a term that doesn't contain any more goals, we have successfully created a proof. [Figure 7].

We now present a complete run of trying to prove that $A \multimap B, A \vdash B$.

We start again by entering the context and the type we are trying to construct. [Figure 8]

```
Enter the context:
f : A -o B, y : A
Enter the intended type of the term:
B
```

FIGURE 8. Setting up $A \multimap B, A \vdash B$

We continue by trying to refine the only goal we have right now, by applying the -oleft rule which will split the context and from part of it prove A and from the other part with a variable of type B included, will try to prove B. [Figure 9]

```
y : A , f : (A ⊸ B)  ⊢ { ?1 } : B
Enter the desired hole #:
1
Hole 1 was selected, with type B
You can use the following variables: y@2 : A, f@1 : (A ⊸ B)
You can use the following meta-variables: 1@4
You can use the following rules: MV, -oleft
Select a rule to be applied:
-oleft
Select the variable to which to apply the rule:
f@1
```

FIGURE 9. The prover suggests that we should apply the -oleft rule, so we do that

---

After we have applied the -oleft, we can now choose which goal to work on. We select goal number 2 and try to provide a term of type A. Since we have $y : A$ in our context, and it has not been used yet, we apply the Id rule to it. [Figure 10]

We now only have the goal number 3 to work on, and its type is B. Since we now have $x : B$ in the resources we are allowed to use (because of the -oleft rule), we apply the Id rule to $x : B$. [Figure 11]

```
y : A , f : (A ⊸ B)  ⊢ let x be (f) ({ ?2 }) in { ?3 } : B
Enter the desired hole #:
2
Hole 2 was selected, with type A
You can use the following variables: y@2 : A
You can use the following meta-variables: 2@8, 3@9
You can use the following rules: MV, Id
Select a rule to be applied:
Id
Select the variable to which to apply the rule:
y@2
```

FIGURE 10. Applying Id rule to $y : A$

```
y : A , f : (A ⊸ B)  ⊢ let x be (f) (y) in { ?3 } : B
Enter the desired hole #:
3
Hole 3 was selected, with type B
You can use the following variables: x@5 : B
You can use the following meta-variables: 3@9
You can use the following rules: MV, Id
Select a rule to be applied:
Id
Select the variable to which to apply the rule:
x@5
```

FIGURE 11. Applying Id rule to $x : B$

Finally we have a complete term without any metavariables in it, and we have used up all of the resources. [Figure 12]

```
y : A , f : (A ⊸ B)  ⊢ let x be (f) (y) in x : B
We are done!
```

FIGURE 12. The final term

---

3. The theory behind Janet

We define a sequent calculus with metavariables in Figure 13. A sequent consists of a metavariable context, a variable context, a resource restriction, a term and a type.

The main challenge when constructing proofs in linear logic is deciding how to split the context when applying rules like tensor right and lollipop left. One approach when proving $A \otimes B$ from Γ, is to use linear contexts, and then split Γ into Γ_1, Γ_2 and prove A from Γ_1 and B from Γ_2. In our implementation we introduce resource multisets in our sequents which are denoted with lowercase Greek letters and represent multisets of variables. A resource multiset α on a context Γ means that we have to use up exactly the resources from Γ that are in α when constructing a term of the given type. Resource multisets have to satisfy restriction equations which are presented in every rule of the calculus. Therefore in our sequent calculus, the variable contexts Γ are always growing when looking at the proofs in a bottom-up manner, and instead of splitting them, we split the resource multisets α.

The metavariable context is a non-linear set of metavariables, where each metavariable is mapped to a context Γ_0, a restriction α_0 and a type A, which means that if we were to plug in a term for a metavariable, we would need to use up exactly the resources α_0 from Γ_0 while constructing a term of type A.

The two rules that deal with resource allocation are $\otimes_R$ and $\multimap_L$. We analyze the $\otimes_R$ rule presented below:

$$\frac{\Delta|\Gamma \vdash_{\alpha_1} e_1 : A \qquad \Delta|\Gamma \vdash_{\alpha_2} e_2 : B \qquad \alpha \cong \alpha_1 \cup \alpha_2}{\Delta|\Gamma \vdash_{\alpha} (e_1, e_2) : A \otimes B} \otimes_R$$

We define $\cong$ as multiset equality. We need to prove $A \otimes B$ from $\Delta|\Gamma$ under α. This means that we need to construct a term of type $A \otimes B$ using only the variables contained in α, and completely using all of them up. All of these variables need to also be contained in Γ. In order to do that, we need to find a split of α into α_1 and α_2 such that we are able to prove A from $\Delta|\Gamma$ under α_1, and B from $\Delta|\Gamma$ under α_2.

Another interesting rule to analyze is the MV rule presented below:

$$\frac{\Delta, u|\Gamma \vdash_{\gamma} \theta : \Gamma_0 \qquad \Delta, u|\Gamma, z : A \vdash_{\beta} e : C \qquad \beta[\frac{\alpha_0[\gamma]}{z}] \cong \alpha}{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_{\alpha} \text{let } z \text{ be } u[\theta] \text{ in } e : C} MV$$

The metavariable rule is the most important rule in our proof assistant as it constructs terms for incomplete parts of the proof which later serve as holes (goals) that the user can refine (substitute new terms in). In order to use a variable to stand for the hole u from $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma$ under α in $e : C$, we need to be able to prove the following:

1. There is a substitution θ (which stands for a list of terms, one for each variable in Γ) under a resource substitution γ, which says that from Γ we can get Γ_0.
2. We can prove $e : C$ from $\Delta, u|\Gamma, z : A$ under β where α is equivalent to substituting the α_0 goal resources under the γ substitution for z in β.

3.1. Two simple proofs.

In the previous section, through Janet we showed how a proof of a linear logic theorem is constructed. We now show two such proofs in our sequent calculus. Since we are interested in how resource allocation works, we display proofs of theorems that use the $\otimes_R$ and $\multimap_L$ rules.

We first prove $A, B \vdash A \otimes B$.

$$\dfrac{\dfrac{x : A \in x : A, y : B}{\Delta | x : A, y : B \vdash_x x : A}\, id \quad \dfrac{y : B \in x : A, y : B}{\Delta | x : A, y : B \vdash_y y : B}\, id \qquad x, y \cong x \cup y}{\Delta | x : A, y : B \vdash_{x,y} (x, y) : A \otimes B}\, \otimes_R$$

Now we show $A \multimap B, A \vdash B$.

$$\dfrac{\dfrac{y : A \in f : A \multimap B, y : A}{\Delta | f : A \multimap B, y : A \vdash_y y : A}\, id \quad \dfrac{x : B \in f : A \multimap B, y : A, x : B}{\Delta | f : A \multimap B, y : A, x : B \vdash_x x : B}\, id \qquad f, y \cong x[\frac{f \cup y}{x}]}{\Delta | f : A \multimap B, y : A \vdash_{f,y} \text{let } x \text{ be } f(y) \text{ in } x : B}\, \multimap_L$$

In order to prove that our system is consistent, we show that the cut rule is admissible. In order to prove that, we need three inversion lemmas which we prove in the next section.

3.2. Inversion Lemmas.

1. If $x : A \otimes B \in \Gamma$ and $\Delta | \Gamma \vdash_\alpha e : C$, then $\exists e'$ s.t. $\Delta | \Gamma, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e' : C$.

2. If $z : 1 \in \Gamma$ and $\Delta | \Gamma \vdash_\alpha e : C$, then $\exists e'$ s.t. $\Delta | \Gamma \vdash_{\alpha[\frac{\cdot}{z}]} e' : C$.

3. If $z : A \oplus B \in \Gamma$ and $\Delta | \Gamma \vdash_\alpha e : C$, then $\exists e', e''$ s.t. $\Delta | \Gamma, x : A \vdash_{\alpha[\frac{x}{z}]} e' : C$ and $\Delta | \Gamma, y : B \vdash_{\alpha[\frac{y}{z}]} e'' : C$.

Proof.

We present a part of the proof of the first lemma. The proofs for the other 2 lemmas follow the same structure.

We have $x : A \otimes B \in \Gamma$.

Base cases:

$\mathbf{1_R}$

$$\Gamma ::= \cdot \mid x : A \mid \Gamma_1 \cup \Gamma_2$$

$$\alpha ::= \cdot \mid x \mid \alpha_1 \cup \alpha_2$$

$$\Delta ::= \cdot \mid u : [\Gamma_0]_{\alpha_0} A \mid \Delta_1 \cup \Delta_2$$

$$\gamma ::= \cdot \mid \gamma,\ \tfrac{\alpha_i}{y_i}$$

$$\frac{}{\Delta|\Gamma \vdash_{\cdot} * : 1}\, 1_R \qquad \frac{\Delta|\Gamma, z : 1 \vdash_{\alpha[\frac{\cdot}{z}]} t : A}{\Delta|\Gamma, z : 1 \vdash_{\alpha} \text{let } * \text{ be } z \text{ in } t : A}\, 1_L \qquad \frac{x : A \in \Gamma}{\Delta|\Gamma \vdash_x x : A}\, \text{id}$$

$$\frac{\Delta|\Gamma \vdash_{\alpha_1} e_1 : A \qquad \Delta|\Gamma \vdash_{\alpha_2} e_2 : B \qquad \alpha \cong \alpha_1 \cup \alpha_2}{\Delta|\Gamma \vdash_{\alpha} (e_1, e_2) : A \otimes B}\, \otimes_R$$

$$\frac{\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e : C}{\Delta|\Gamma, x : A \otimes B \vdash_{\alpha} \text{let } (x_1, x_2) \text{ be } x \text{ in } e : C}\, \otimes_L$$

$$\frac{\Delta|\Gamma, x : A \vdash_{\alpha \cup x} t : B}{\Delta|\Gamma \vdash_{\alpha} \lambda x.t : A \multimap B}\, \multimap_R$$

$$\frac{\Delta|\Gamma, f : A \multimap B \vdash_{\alpha_1} t : A \qquad \Delta|\Gamma, f : A \multimap B, x : B \vdash_{\alpha_2} e : C \qquad \alpha \cong \alpha_2[\frac{f \cup \alpha_1}{x}]}{\Delta|\Gamma, f : A \multimap B \vdash_{\alpha} \text{let } x \text{ be } f(t) \text{ in } e : C}\, \multimap_L$$

$$\frac{\Delta|\Gamma \vdash_{\alpha} t : A}{\Delta|\Gamma \vdash_{\alpha} inl(t) : A \oplus B}\, \oplus_{R_1} \qquad \frac{\Delta|\Gamma \vdash_{\alpha} t : B}{\Delta|\Gamma \vdash_{\alpha} inr(t) : A \oplus B}\, \oplus_{R_2}$$

$$\frac{\Delta|\Gamma, z : A \oplus B, x : A \vdash_{\alpha[\frac{x}{z}]} e_1 : C \qquad \Delta|\Gamma, z : A \oplus B, y : B \vdash_{\alpha[\frac{y}{z}]} e_2 : C}{\Delta|\Gamma, z : A \oplus B \vdash_{\alpha} \text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2 : C}\, \oplus_L$$

$$\frac{\Delta|\Gamma \vdash_{\alpha} e_1 : A \qquad \Delta|\Gamma \vdash_{\alpha} e_2 : B}{\Delta|\Gamma \vdash_{\alpha} < e_1, e_2 > : A \,\&\, B}\, \&_R$$

$$\frac{\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\alpha'} e : C \qquad \alpha \cong \alpha'[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\alpha} \text{let } < x, \_ > \text{ be } z \text{ in } e : C}\, \&_{L_1}$$

$$\frac{\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\alpha'} e : C \qquad \alpha \cong \alpha'[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\alpha} \text{let } < \_, x > \text{ be } z \text{ in } e : C}\, \&_{L_2}$$

$$\frac{\Delta, u|\Gamma \vdash_{\gamma} \theta : \Gamma_0 \qquad \Delta, u|\Gamma, z : A \vdash_{\beta} e : C \qquad \beta[\frac{\alpha_0[\gamma]}{z}] \cong \alpha}{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_{\alpha} \text{let } z \text{ be } u[\theta] \text{ in } e : C}\, MV$$

FIGURE 13. Linear Logic Sequent Calculus with Metavariables

$$\frac{}{\Delta|\Gamma \vdash_{\cdot} * : 1}\, 1_R$$

Then since $\cdot \cong \cdot[\frac{x_1 \cup x_2}{x}]$ and we can weaken Γ we have

$$\frac{}{\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{\cdot[\frac{x_1 \cup x_2}{x}]} * : 1}\, 1_R$$

so we are done.

id

We have 2 cases: when the variable to which id is applied is not x

$$\frac{y : C \in \Gamma}{\Delta|\Gamma \vdash_y y : C}\, \text{id}$$

Then since $y \neq x$, $y \cong y[\frac{x_1 \cup x_2}{x}]$, so by just weakening Γ, we get:

$$\frac{y : C \in \Gamma, x_1 : A, x_2 : B}{\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{y[\frac{x_1 \cup x_2}{x}]} y : C}\, \text{id}$$

and the case where the variable to which id is applied is x

$$\frac{x : A \otimes B \in \Gamma}{\Delta|\Gamma \vdash_x x : A \otimes B}\, \text{id}$$

Then we can construct the following term:

$$\frac{\dfrac{x_1 : A \in \Gamma, x_1 : A, x_2 : B}{\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{x_1} x_1 : A}\, \text{id} \quad \dfrac{x_2 : B \in \Gamma, x_1 : A, x_2 : B}{\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{x_2} x_2 : B}\, \text{id} \quad x[\frac{x_1 \cup x_2}{x}] \cong x_1 \cup x_2}{\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{x[\frac{x_1 \cup x_2}{x}]} (x_1, x_2) : A \otimes B}\, \otimes_R$$

Inductive cases:

$\otimes_{\boldsymbol{R}}$

$$\frac{\Delta|\Gamma \vdash_{\alpha_1} e_1 : A \qquad \Delta|\Gamma \vdash_{\alpha_2} e_2 : B \qquad \alpha \cong \alpha_1 \cup \alpha_2}{\Delta|\Gamma \vdash_{\alpha} (e_1, e_2) : A \otimes B} \otimes_R$$

By applying IH twice we get $\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{\alpha_1[\frac{x_1 \cup x_2}{x}]} e_1' : A$ and $\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{\alpha_2[\frac{x_1 \cup x_2}{x}]} e_2' : B$. Now $\alpha_1[\frac{x_1 \cup x_2}{x}] \cup \alpha_2[\frac{x_1 \cup x_2}{x}] \cong (\alpha_1 \cup \alpha_2)[\frac{x_1 \cup x_2}{x}] \cong \alpha[\frac{x_1 \cup x_2}{x}]$.

$$\frac{\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{\alpha_1[\frac{x_1 \cup x_2}{x}]} e_1' : A \qquad \Delta|\Gamma, x_1 : A, x_2 : B \vdash_{\alpha_2[\frac{x_1 \cup x_2}{x}]} e_2' : B \qquad \alpha[\frac{x_1 \cup x_2}{x}] \cong \alpha_1[\frac{x_1 \cup x_2}{x}] \cup \alpha_2[\frac{x_1 \cup x_2}{x}]}{\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} (e_1', e_2') : A \otimes B} \otimes_R$$

$\otimes_{\boldsymbol{L}}$

Again we deal with two cases, one when the rule is not applied to the $x : A \otimes B$ variable:

$$\frac{\Delta|\Gamma, y : C \otimes D, y_1 : C, y_2 : D \vdash_{\alpha[\frac{y_1 \cup y_2}{y}]} e : E}{\Delta|\Gamma, y : C \otimes D \vdash_{\alpha} \text{let } (y_1, y_2) \text{ be } y \text{ in } e : E} \otimes_L$$

By applying IH we get $\Delta|\Gamma, y : C \otimes D, y_1 : C, y_2 : D, x_1 : A, x_2 : B \vdash_{(\alpha[\frac{y_1 \cup y_2}{y}])[\frac{x_1 \cup x_2}{x}]} e' : E$, and since $y \neq x$, we have $(\alpha[\frac{y_1 \cup y_2}{y}])[\frac{x_1 \cup x_2}{x}] \cong (\alpha[\frac{x_1 \cup x_2}{x}])[\frac{y_1 \cup y_2}{y}]$. So by applying $\otimes_L$ again, we get

$$\frac{\Delta|\Gamma, y : C \otimes D, y_1 : C, y_2 : D, x_1 : A, x_2 : B \vdash_{(\alpha[\frac{y_1 \cup y_2}{y}])[\frac{x_1 \cup x_2}{x}]} e' : E}{\Delta|\Gamma, y : C \otimes D, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} \text{let } (y_1, y_2) \text{ be } y \text{ in } e' : E} \otimes_L$$

In the case that the rule is applied to the $x : A \otimes B$ variable we have:

$$\frac{\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e : C}{\Delta|\Gamma \vdash_\alpha \text{let } (x_1, x_2) \text{ be } x \text{ in } e : C} \otimes_L$$

so since we already have $\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e : C$, we are done.

The rest of the proof is left as an exercise for the reader.

□

3.3. Admissibility of cut.

Theorem:

1. If $\Delta|\Gamma \vdash_\alpha e' : A$ and $\Delta|\Gamma, x : A \vdash_\beta e : B$ then $\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} e[\frac{e'}{x}] : B$. (cut)
2. If $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : A$ and $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_\beta e : B$ then $\Delta|\Gamma \vdash_\beta e[\frac{e'}{u}] : B$. (cut!)

Proof.

1. We write l(left) for the derivation of A and r(right) for the derivation of B. We proceed by induction on r by first showing what happens in the case that the last step of the derivation of r is a right rule. We then consider the cases where the last rule applied in the derivation of r is a left rule. In those cases, the last rule applied in the derivation of l is either the matching right rule, or any other left rule.

Base cases:

$\mathbf{1_R}$

$$\frac{\Delta|\Gamma \vdash_\alpha e' : A \qquad \dfrac{}{\Delta|\Gamma, x : A \vdash_\cdot * : 1}\, 1_R}{\Delta|\Gamma \vdash_{\cdot[\frac{\alpha}{x}]} *[\frac{e'}{x}] : 1}\, cut$$

Then since $\cdot \cong \cdot[\frac{\alpha}{x}]$ from 1_R we have

$$\frac{}{\Delta|\Gamma \vdash_{\cdot[\frac{\alpha}{x}]} *[\frac{e'}{x}] : 1}\, 1_R$$

so we are done.

id

We get 2 cases: if the variable being cut is the one to which the *id* rule is applied:

$$\frac{\Delta|\Gamma \vdash_\alpha e' : A \qquad \dfrac{x : A \in \Gamma}{\Delta|\Gamma, x : A \vdash_x x : A}\,\text{id}}{\Delta|\Gamma \vdash_{x[\frac{\alpha}{x}]} x[\frac{e'}{x}] : A}\, cut$$

Then since $x[\frac{\alpha}{x}] \cong \alpha$ from l we have $\Delta|\Gamma \vdash_\alpha e' : A$, so we are done.

and the case where the cut variable isn't the one to which the *id* rule is applied:

$$\frac{\Delta|\Gamma, y : B \vdash_\alpha e' : A \qquad \dfrac{y : B \in \Gamma}{\Delta|\Gamma, x : A, y : B \vdash_y y : B}\,\text{id}}{\Delta|\Gamma, y : B \vdash_{y[\frac{\alpha}{x}]} y[\frac{e'}{x}] : B}\, cut$$

Then since $y[\frac{\alpha}{x}] \cong y$ from *id* we have $\Delta|\Gamma, y : B \vdash_y y : B$, so we are done.

Inductive cases:

We will first do all of the right rules.

$\otimes_{\boldsymbol{R}}$

$$\frac{\Delta|\Gamma \vdash_\alpha e' : C \qquad \dfrac{\Delta|\Gamma, x : C \vdash_{\beta_1} e_1 : A \qquad \Delta|\Gamma, x : C \vdash_{\beta_2} e_2 : B \qquad \beta \cong \beta_1 \cup \beta_2}{\Delta|\Gamma, x : C \vdash_\beta (e_1, e_2) : A \otimes B}\,\otimes_R}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} (e_1, e_2)[\frac{e'}{x}] : A \otimes B}\, cut$$

Now we can cut $\Delta|\Gamma \vdash_\alpha e' : C$ and $\Delta|\Gamma, x : C \vdash_{\beta_1} e_1 : A$ to get $\Delta|\Gamma \vdash_{\beta_1[\frac{\alpha}{x}]} e_1[\frac{e'}{x}] : A$, as well as $\Delta|\Gamma \vdash_\alpha e' : C$ and $\Delta|\Gamma, x : C \vdash_{\beta_2} e_2 : B$ to get $\Delta|\Gamma \vdash_{\beta_2[\frac{\alpha}{x}]}$

$e_2[\frac{e'}{x}] : B$. Furthermore applying the substitution $[\frac{\alpha}{x}]$ to both sides of $\beta \cong \beta_1 \cup \beta_2$, we get $\beta[\frac{\alpha}{x}] \cong (\beta_1 \cup \beta_2)[\frac{\alpha}{x}]$ which is equivalent to $\beta[\frac{\alpha}{x}] \cong \beta_1[\frac{\alpha}{x}] \cup \beta_2[\frac{\alpha}{x}]$. Then applying $\otimes_R$ we get

$$\frac{\Delta|\Gamma \vdash_{\beta_1[\frac{\alpha}{x}]} e_1[\frac{e'}{x}] : A \qquad \Delta|\Gamma \vdash_{\beta_2[\frac{\alpha}{x}]} e_2[\frac{e'}{x}] : B \qquad \beta[\frac{\alpha}{x}] \cong \beta_1[\frac{\alpha}{x}] \cup \beta_2[\frac{\alpha}{x}]}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} (e_1[\frac{e'}{x}], e_2[\frac{e'}{x}]) : A \otimes B} \otimes_R$$

$\multimap_R$

$$\frac{\Delta|\Gamma \vdash_\alpha e' : C \qquad \dfrac{\Delta|\Gamma, x : A, y : C \vdash_{\beta \cup x} e : B}{\Delta|\Gamma, y : C \vdash_\beta \lambda x.e : A \multimap B} \multimap_R}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{y}]} (\lambda x.t)[\frac{e'}{y}] : A \otimes B} \, cut$$

Now we can cut $\Delta|\Gamma \vdash_\alpha e' : C$ and $\Delta|\Gamma, x : A, y : C \vdash_{\beta \cup x} e : B$ to get $\Delta|\Gamma, x : A \vdash_{(\beta \cup x)[\frac{\alpha}{y}]} e[\frac{e'}{y}] : B$. Then since $y \neq x$, $(\beta \cup x)[\frac{\alpha}{y}] \cong (\beta[\frac{\alpha}{y}]) \cup x$, so by applying $\multimap_R$ we get

$$\frac{\Delta|\Gamma, x : A \vdash_{(\beta[\frac{\alpha}{y}]) \cup x} e[\frac{e'}{y}] : B}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{y}]} \lambda x.(t[\frac{e'}{y}]) : A \multimap B} \multimap_R$$

$\oplus_{R_1}$

$$\frac{\Delta|\Gamma \vdash_\alpha e' : C \qquad \dfrac{\Delta|\Gamma, x : C \vdash_\beta e : A}{\Delta|\Gamma, x : C \vdash_\beta inl(e) : A \oplus B} \oplus_{R_1}}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} inl(e)[\frac{e'}{x}] : A \oplus B} \, cut$$

Now we can cut $\Delta|\Gamma \vdash_\alpha e' : C$ and $\Delta|\Gamma, x : C \vdash_\beta e : A$ to get $\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} e[\frac{e'}{x}] : A$. Then by applying $\oplus_{R_1}$ we get

$$\frac{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} e[\frac{e'}{x}] : A}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} inl(e[\frac{e'}{x}]) : A \oplus B} \oplus_{R_1}$$

$\oplus_{R_2}$

$$\frac{\Delta|\Gamma \vdash_\alpha e' : C \qquad \dfrac{\Delta|\Gamma, x : C \vdash_\beta e : B}{\Delta|\Gamma, x : C \vdash_\beta inr(e) : A \oplus B}\oplus_{R_2}}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} inr(e)[\frac{e'}{x}] : A \oplus B}\, cut$$

Now we can cut $\Delta|\Gamma \vdash_\alpha e' : C$ and $\Delta|\Gamma, x : C \vdash_\beta e : B$ to get $\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} e[\frac{e'}{x}] : B$. Then by applying $\oplus_{R_2}$ we get

$$\frac{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} e[\frac{e'}{x}] : B}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} inr(e[\frac{e'}{x}]) : A \oplus B}\oplus_{R_2}$$

$\&_R$

$$\frac{\Delta|\Gamma \vdash_\alpha e' : C \qquad \dfrac{\Delta|\Gamma, x : C \vdash_\beta e_1 : A \qquad \Delta|\Gamma, x : C \vdash_\beta e_2 : B}{\Delta|\Gamma, x : C \vdash_\beta < e_1, e_2 > : A \,\&\, B}\&_R}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} < e_1, e_2 > [\frac{e'}{x}] : A \,\&\, B}\, cut$$

Now we can cut $\Delta|\Gamma \vdash_\alpha e' : C$ and $\Delta|\Gamma, x : C \vdash_\beta e_1 : A$ to get $\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} e_1[\frac{e'}{x}] : A$, as well as $\Delta|\Gamma \vdash_\alpha e' : C$ and $\Delta|\Gamma, x : C \vdash_\beta e_2 : B$ to get $\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} e_2[\frac{e'}{x}] : B$. Then by applying $\&_R$ we get

$$\frac{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} e_1[\frac{e'}{x}] : A \qquad \Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} e_2[\frac{e'}{x}] : B}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} < e_1[\frac{e'}{x}], e_2[\frac{e'}{x}] > : A \,\&\, B}\&_R$$

MV

$$\frac{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_\alpha e' : B \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, x : B \vdash_\gamma \theta : \Gamma_0 \qquad \Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, z : A, x : B \vdash_\delta e : C \qquad \delta[\frac{\alpha_0[\gamma]}{z}] \cong \beta}{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, x : B \vdash_\beta \text{let } z \text{ be } u[\theta] \text{ in } e : C}\, MV}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} (\text{let } z \text{ be } u[\theta] \text{ in } e)[\frac{e'}{x}] : C}\, cut$$

Now we can weaken and then cut $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_\alpha e' : B$ and $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, z : A, x : B \vdash_\delta e : C$ to get $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, z : A \vdash_{\delta[\frac{\alpha}{x}]} e[\frac{e'}{x}] : C$, as well as $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_\alpha e' : B$ and $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, x : B \vdash_\gamma \theta : \Gamma_0$, by iterating the inductive hypothesis for each term in θ, to get $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_{\gamma[\frac{\alpha}{x}]} \theta[\frac{e'}{x}] : \Gamma_0$. Then by applying MV we get

$$\frac{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_{\gamma[\frac{\alpha}{x}]} \theta[\frac{e'}{x}] : \Gamma_0 \qquad \Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, z : A \vdash_{\delta[\frac{\alpha}{x}]} e[\frac{e'}{x}] : C \qquad (\delta[\frac{\alpha}{x}])[\frac{\alpha_0[\gamma[\frac{\alpha}{x}]]}{z}] \cong \beta[\frac{\alpha}{x}]}{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} \text{let } z \text{ be } u[\theta[\frac{e'}{x}]] \text{ in } e[\frac{e'}{x}] : C} MV$$

If the last rule in the derivation of r(right) is a left rule, we have to consider 2 separate cases. One where the cut variable is not the variable on which the last rule in the derivation of l(left) is applied, and one where it is. We will first consider the cases where the variable cut is not used in the last rule of the derivation of r.

1_L

$$\frac{\Delta|\Gamma \vdash_\alpha e' : B \qquad \dfrac{\Delta|\Gamma, z : 1, x : B \vdash_{\beta[\frac{1}{z}]} e : A}{\Delta|\Gamma, z : 1, x : B \vdash_\beta \text{let } 1 \text{ be } z \text{ in } e : A} 1_L}{\Delta|\Gamma, z : 1 \vdash_{\beta[\frac{\alpha}{x}]} (\text{let } 1 \text{ be } z \text{ in } e)[\frac{e'}{x}] : A} cut$$

Now we can weaken and cut $\Delta|\Gamma \vdash_\alpha e' : B$ and $\Delta|\Gamma, z : 1, x : B \vdash_{\beta[\frac{1}{z}]} e : A$ to get $\Delta|\Gamma, z : 1 \vdash_{(\beta[\frac{1}{z}])[\frac{\alpha}{x}]} e[\frac{e'}{x}] : A$. We also have $(\beta[\frac{1}{z}])[\frac{\alpha}{x}] \cong (\beta[\frac{\alpha}{x}])[\frac{1}{z}]$, since $z \notin \alpha$. So by applying 1_L we get

$$\frac{\Delta|\Gamma, z : 1 \vdash_{(\beta[\frac{\alpha}{x}])[\frac{1}{z}]} e[\frac{e'}{x}] : A}{\Delta|\Gamma, z : 1 \vdash_{\beta[\frac{\alpha}{x}]} \text{let } 1 \text{ be } z \text{ in } e[\frac{e'}{x}] : A} 1_L$$

$\otimes_{\boldsymbol{L}}$

$$\dfrac{\Delta|\Gamma \vdash_{\alpha} e' : D \qquad \dfrac{\Delta|\Gamma, y : D, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\beta[\frac{x_1 \cup x_2}{x}]} e : C}{\Delta|\Gamma, y : D, x : A \otimes B \vdash_{\beta} \text{let } (x_1, x_2) \text{ be } x \text{ in } e : C} \otimes_L}{\Delta|\Gamma, x : A \otimes B \vdash_{\beta[\frac{\alpha}{y}]} (\text{let } (x_1, x_2) \text{ be } x \text{ in } e)[\frac{e'}{y}] : C} \, cut$$

Now we can weaken and then cut $\Delta|\Gamma \vdash_{\alpha} e' : D$ and $\Delta|\Gamma, y : D, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\beta[\frac{x_1 \cup x_2}{x}]} e : C$ to get $\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{(\beta[\frac{x_1 \cup x_2}{x}])[\frac{\alpha}{y}]} e[\frac{e'}{x}] : C$. Now since $(\beta[\frac{x_1 \cup x_2}{x}])[\frac{\alpha}{y}] \cong (\beta[\frac{\alpha}{y}])[\frac{x_1 \cup x_2}{x}]$ by applying $\otimes_L$ we get

$$\dfrac{\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{(\beta[\frac{\alpha}{y}])[\frac{x_1 \cup x_2}{x}]} e[\frac{e'}{y}] : C}{\Delta|\Gamma, x : A \otimes B \vdash_{\beta[\frac{\alpha}{y}]} \text{let } (x_1, x_2) \text{ be } x \text{ in } e[\frac{e'}{y}] : C} \otimes_L$$

$\multimap_{\boldsymbol{L}}$

$$\dfrac{\Delta|\Gamma \vdash_{\alpha} e' : D \qquad \dfrac{\Delta|\Gamma, y : D, f : A \multimap B \vdash_{\beta_1} t : A \qquad \Delta|\Gamma, y : D, f : A \multimap B, x : B \vdash_{\beta_2} e : C \qquad \beta \cong \beta_2[\frac{f \cup \beta_1}{x}]}{\Delta|\Gamma, y : D, f : A \multimap B \vdash_{\beta} \text{let } x \text{ be } f(t) \text{ in } e : C} \multimap_L}{\Delta|\Gamma, f : A \multimap B \vdash_{\beta[\frac{\alpha}{y}]} (\text{let } x \text{ be } f(t) \text{ in } e)[\frac{e'}{y}] : C} \, cut$$

Now we can weaken and cut $\Delta|\Gamma \vdash_{\alpha} e' : D$ and $\Delta|\Gamma, y : D, f : A \multimap B \vdash_{\beta_1} t : A$ to get $\Delta|\Gamma, f : A \multimap B \vdash_{\beta_1[\frac{\alpha}{y}]} t[\frac{e'}{y}] : A$, as well as $\Delta|\Gamma \vdash_{\alpha} e' : D$ and $\Delta|\Gamma, y : D, f : A \multimap B, x : B \vdash_{\beta_2} e : C$ to get $\Delta|\Gamma, f : A \multimap B, x : B \vdash_{\beta_2[\frac{\alpha}{y}]} e[\frac{e'}{y}] : C$. Then by applying $\multimap_L$ we get

$$\dfrac{\Delta|\Gamma, f : A \multimap B \vdash_{\beta_1[\frac{\alpha}{y}]} t[\frac{e'}{y}] : A \qquad \Delta|\Gamma, f : A \multimap B, x : B \vdash_{\beta_2[\frac{\alpha}{y}]} e[\frac{e'}{y}] : C \qquad \beta[\frac{\alpha}{y}] \cong (\beta_2[\frac{\alpha}{y}])[\frac{f \cup (\beta_1[\frac{\alpha}{y}])}{x}]}{\Delta|\Gamma, f : A \multimap B \vdash_{\beta[\frac{\alpha}{y}]} \text{let } x \text{ be } f(t[\frac{e'}{y}]) \text{ in } e[\frac{e'}{y}] : C} \multimap_L$$

$\oplus_{\boldsymbol{L}}$

$$\dfrac{\Delta|\Gamma \vdash_{\alpha} e' : D \qquad \dfrac{\Delta|\Gamma, w : D, z : A \oplus B, x : A \vdash_{\beta[\frac{x}{z}]} e_1 : C \qquad \Delta|\Gamma, w : D, z : A \oplus B, y : B \vdash_{\beta[\frac{y}{z}]} e_2 : C}{\Delta|\Gamma, w : D, z : A \oplus B \vdash_{\beta} \text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2 : C} \oplus_L}{\Delta|\Gamma, z : A \oplus B \vdash_{\beta[\frac{\alpha}{w}]} (\text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2)[\frac{e'}{w}] : C} \, cut$$

Now we can weaken and cut $\Delta|\Gamma \vdash_\alpha e' : D$ and $\Delta|\Gamma, w : D, z : A \oplus B, x : A \vdash_{\beta[\frac{x}{z}]} e_1 : C$ to get $\Delta|\Gamma, z : A \oplus B, x : A \vdash_{(\beta[\frac{x}{z}])[\frac{\alpha}{w}]} e_1[\frac{e'}{w}] : C$, as well as $\Delta|\Gamma \vdash_\alpha e' : D$ and $\Delta|\Gamma, w : D, z : A \oplus B, y : B \vdash_{\beta[\frac{y}{z}]} e_2 : C$ to get $\Delta|\Gamma, z : A \oplus B, y : B \vdash_{(\beta[\frac{y}{z}])[\frac{\alpha}{w}]} e_2[\frac{e'}{w}] : C$. Since $(\beta[\frac{x}{z}])[\frac{\alpha}{w}] \cong (\beta[\frac{\alpha}{w}])[\frac{x}{z}]$ and $(\beta[\frac{y}{z}])[\frac{\alpha}{w}] \cong (\beta[\frac{\alpha}{w}])[\frac{y}{z}]$ we can apply $\oplus_L$ and get

$$\frac{\Delta|\Gamma, z : A \oplus B, x : A \vdash_{(\beta[\frac{\alpha}{w}])[\frac{x}{z}]} e_1[\frac{e'}{w}] : C \qquad \Delta|\Gamma, z : A \oplus B, y : B \vdash_{(\beta[\frac{\alpha}{w}])[\frac{y}{z}]} e_2[\frac{e'}{w}] : C}{\Delta|\Gamma, z : A \oplus B \vdash_{\beta[\frac{\alpha}{w}]} \text{case } z \text{ of } inl(x) => e_1[\frac{e'}{w}], inr(y) => e_2[\frac{e'}{w}] : C} \oplus_L$$

$\&_{L_1}$

$$\frac{\Delta|\Gamma, z : A \,\&\, B \vdash_\alpha e' : D \qquad \dfrac{\Delta|\Gamma, y : D, z : A \,\&\, B, x : A \vdash_{\beta'} e : C \qquad \beta \cong \beta'[\frac{z}{x}]}{\Delta|\Gamma, y : D, z : A \,\&\, B \vdash_\beta \text{let } < x, \_ > \text{ be } z \text{ in } e : C} \,\&_{L_1}}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\beta[\frac{\alpha}{y}]} (\text{let } < x, \_ > \text{ be } z \text{ in } e)[\frac{e'}{y}] : C} \, cut$$

Now we can weaken and cut $\Delta|\Gamma, z : A \,\&\, B \vdash_\alpha e' : D$ and $\Delta|\Gamma, y : D, z : A \,\&\, B, x : A \vdash_{\beta'} e : C$ to get $\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\beta'[\frac{\alpha}{y}]} e[\frac{e'}{y}] : C$. We have $(\beta'[\frac{\alpha}{y}])[\frac{z}{x}] \cong (\beta'[\frac{z}{x}])[\frac{\alpha}{y}] \cong \beta[\frac{\alpha}{y}]$ because $x \notin \alpha$, so by applying $\&_{L_1}$ we get

$$\frac{\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\beta'[\frac{\alpha}{y}]} e[\frac{e'}{y}] : C \qquad \beta[\frac{\alpha}{y}] \cong (\beta'[\frac{\alpha}{y}])[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\beta[\frac{\alpha}{y}]} \text{let } < x, \_ > \text{ be } z \text{ in } e[\frac{e'}{y}] : C} \,\&_{L_1}$$

$\&_{L_2}$

$$\frac{\Delta|\Gamma, z : A \,\&\, B \vdash_\alpha e' : D \qquad \dfrac{\Delta|\Gamma, y : D, z : A \,\&\, B, x : B \vdash_{\beta'} e : C \qquad \beta \cong \beta'[\frac{z}{x}]}{\Delta|\Gamma, y : D, z : A \,\&\, B \vdash_\beta \text{let } < \_, x > \text{ be } z \text{ in } e : C} \,\&_{L_2}}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\beta[\frac{\alpha}{y}]} (\text{let } < \_, x > \text{ be } z \text{ in } e)[\frac{e'}{y}] : C} \, cut$$

Now we can weaken and cut $\Delta|\Gamma, z : A \,\&\, B \vdash_\alpha e' : D$ and $\Delta|\Gamma, y : D, z : A \,\&\, B, x : B \vdash_{\beta'} e : C$ to get $\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\beta'[\frac{\alpha}{y}]} e[\frac{e'}{y}] : C$. We have $(\beta'[\frac{\alpha}{y}])[\frac{z}{x}] \cong (\beta'[\frac{z}{x}])[\frac{\alpha}{y}] \cong \beta[\frac{\alpha}{y}]$ because $x \notin \alpha$, so by applying $\&_{L_2}$ we get

$$\frac{\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\beta'[\frac{\alpha}{y}]} e[\frac{e'}{y}] : C \qquad \beta[\frac{\alpha}{y}] \cong (\beta'[\frac{\alpha}{y}])[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\beta[\frac{\alpha}{y}]} \text{let } < \_, x > \text{ be } z \text{ in } e[\frac{e'}{y}] : C}\,\&_{L_2}$$

Now we need to consider the case where the cut variable is used in the last step of the derivation of r(right). In this case the last step of the derivation of l, has to either be the right rule for the same operator, or any other left rule. Again we split in 2 cases: first we will prove the cases where the derivation of l ends in the right rule for the same operator, and then we will prove the cut theorem for all the left rules as last steps in the derivation of l.

$\mathbf{1_R}$ and $\mathbf{1_L}$

$$1_R\,\frac{\dfrac{}{\Delta|\Gamma \vdash_{\cdot} * : 1} \qquad \dfrac{\Delta|\Gamma, z : 1 \vdash_{\beta[\frac{\cdot}{z}]} e : A}{\Delta|\Gamma, z : 1 \vdash_{\beta} \text{let } * \text{ be } z \text{ in } e : A}\,1_L}{\Delta|\Gamma \vdash_{\beta[\frac{\cdot}{z}]} (\text{let } * \text{ be } z \text{ in } e)[\frac{*}{z}] : A}\,cut$$

Now we can cut $\Delta|\Gamma \vdash_{\cdot} * : 1$ and $\Delta|\Gamma, z : 1 \vdash_{\beta[\frac{\cdot}{z}]} e : A$ to get $\Delta|\Gamma \vdash_{(\beta[\frac{\cdot}{z}])[\frac{\cdot}{z}]} e[\frac{*}{z}] : A$. But since $(\beta[\frac{\cdot}{z}])[\frac{\cdot}{z}] \cong \beta[\frac{\cdot}{z}]$ we have $\Delta|\Gamma \vdash_{\beta[\frac{\cdot}{z}]} e[\frac{*}{z}] : A$.

$\otimes_{\mathbf{R}}$ and $\otimes_{\mathbf{L}}$

$$\otimes_R\,\frac{\dfrac{\Delta|\Gamma \vdash_{\alpha_1} e_1 : A \qquad \Delta|\Gamma \vdash_{\alpha_2} e_2 : B \qquad \alpha \cong \alpha_1 \cup \alpha_2}{\Delta|\Gamma \vdash_{\alpha} (e_1, e_2) : A \otimes B} \qquad \dfrac{\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\beta[\frac{x_1 \cup x_2}{x}]} e : C}{\Delta|\Gamma, x : A \otimes B \vdash_{\beta} \text{let } (x_1, x_2) \text{ be } x \text{ in } e : C}\,\otimes_L}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} (\text{let } (x_1, x_2) \text{ be } x \text{ in } e)[\frac{(e_1, e_2)}{x}] : C}\,cut$$

Now we can weaken and then cut $\Delta|\Gamma \vdash_{\alpha} (e_1, e_2) : A \otimes B$ into $\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\beta[\frac{x_1 \cup x_2}{x}]} e : C$ to get $\Delta|\Gamma, x_1 : A, x_2 : B \vdash_{(\beta[\frac{x_1 \cup x_2}{x}])[\frac{\alpha}{x}]} e[\frac{(e_1, e_2)}{x}] : C$. Now weaken and cut $\Delta|\Gamma \vdash_{\alpha_1} e_1 : A$ into this to get $\Delta|\Gamma, x_2 :$

$B \vdash_{((\beta[\frac{x_1 \cup x_2}{x}])[\frac{\alpha}{x}])[\frac{\alpha_1}{x_1}]} (e[\frac{(e_1,e_2)}{x}])[\frac{e_1}{x_1}] : C$. Finally weaken and cut $\Delta|\Gamma \vdash_{\alpha_2} e_2 : B$ into the previous result to get $\Delta|\Gamma \vdash_{(((\beta[\frac{x_1 \cup x_2}{x}])[\frac{\alpha}{x}])[\frac{\alpha_1}{x_1}])[\frac{\alpha_2}{x_2}]} ((e[\frac{(e_1,e_2)}{x}])[\frac{e_1}{x_1}])[\frac{e_2}{x_2}] : C$. Now since $(((\beta[\frac{x_1 \cup x_2}{x}])[\frac{\alpha}{x}])[\frac{\alpha_1}{x_1}])[\frac{\alpha_2}{x_2}] \cong (\beta[\frac{\alpha_1 \cup \alpha_2}{x}])[\frac{\alpha}{x}]$ and since $\alpha \cong \alpha_1 \cup \alpha_2$, thats equivalent to $(\beta[\frac{\alpha}{x}])[\frac{\alpha}{x}]$, and now since $x \notin \alpha$ that is equivalent to just $\beta[\frac{\alpha}{x}]$, so

$$\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} ((e[\frac{(e_1,e_2)}{x}])[\frac{e_1}{x_1}])[\frac{e_2}{x_2}] : C$$

$\multimap_R$ and **$\multimap_L$**

$$\multimap_R \frac{\dfrac{\Delta|\Gamma, z : A \vdash_{\alpha \cup z} e' : B}{\Delta|\Gamma \vdash_{\alpha} \lambda z.e' : A \multimap B} \quad \dfrac{\Delta|\Gamma, f : A \multimap B \vdash_{\beta_1} t : A \qquad \Delta|\Gamma, f : A \multimap B, x : B \vdash_{\beta_2} e : C \qquad \beta \cong \beta_2[\frac{f \cup \beta_1}{x}]}{\Delta|\Gamma, f : A \multimap B \vdash_{\beta} \text{let } x \text{ be } f(t) \text{ in } e : C} \multimap_L}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{f}]} (\text{let } x \text{ be } f(t) \text{ in } e)[\frac{\lambda z.e'}{f}] : C} \, cut$$

Now we can cut $\Delta|\Gamma \vdash_{\alpha} \lambda z.e' : A \multimap B$ into $\Delta|\Gamma, f : A \multimap B \vdash_{\beta_1} t : A$ to get $\Delta|\Gamma \vdash_{\beta_1[\frac{\alpha}{f}]} t[\frac{\lambda z.e'}{f}] : A$. Furthermore, weaken and cut $\Delta|\Gamma \vdash_{\alpha} \lambda z.e' : A \multimap B$ into $\Delta|\Gamma, f : A \multimap B, x : B \vdash_{\beta_2} e : C$ to get $\Delta|\Gamma, x : B \vdash_{\beta_2[\frac{\alpha}{f}]} e[\frac{\lambda z.e'}{f}] : C$. Now cut $\Delta|\Gamma \vdash_{\beta_1[\frac{\alpha}{f}]} t[\frac{\lambda z.e'}{f}] : A$ into $\Delta|\Gamma, z : A \vdash_{\alpha \cup z} e' : B$ to get $\Delta|\Gamma \vdash_{(\alpha \cup z)[\frac{\beta_1[\frac{\alpha}{f}]}{z}]} e'[\frac{t[\frac{\lambda z.e'}{f}]}{z}] : B$. Now since $z \notin \alpha$ we get $\Delta|\Gamma \vdash_{\alpha \cup \beta_1[\frac{\alpha}{f}]} e'[\frac{t[\frac{\lambda z.e'}{f}]}{z}] : B$. Finally cut this result into $\Delta|\Gamma, x : B \vdash_{\beta_2[\frac{\alpha}{f}]} e[\frac{\lambda z.e'}{f}] : C$ to get $\Delta|\Gamma \vdash_{(\beta_2[\frac{\alpha}{f}])[\frac{\alpha \cup \beta_1[\frac{\alpha}{f}]}{x}]} (e[\frac{\lambda z.e'}{f}])[\frac{e'[\frac{t[\frac{\lambda z.e'}{f}]}{z}]}{x}] : C$.

Now we have $\beta[\frac{\alpha}{f}] \cong (\beta_2[\frac{f \cup \beta_1}{x}])[\frac{\alpha}{f}] \cong (\beta_2[\frac{\alpha}{f}])[\frac{\alpha \cup \beta_1[\frac{\alpha}{f}]}{x}]$. So we are done.

$\oplus_{R_1}$ and **$\oplus_L$**

$$\oplus_{R_1} \frac{\dfrac{\Delta|\Gamma \vdash_{\alpha} e' : A}{\Delta|\Gamma \vdash_{\alpha} inl(e') : A \oplus B} \quad \dfrac{\Delta|\Gamma, z : A \oplus B, x : A \vdash_{\beta[\frac{x}{z}]} e_1 : C \qquad \Delta|\Gamma, z : A \oplus B, y : B \vdash_{\beta[\frac{y}{z}]} e_2 : C}{\Delta|\Gamma, z : A \oplus B \vdash_{\beta} \text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2 : C} \oplus_L}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{z}]} (\text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2)[\frac{inl(e')}{z}] : C} \, cut$$

Now we can weaken and cut $\Delta|\Gamma \vdash_\alpha inl(e') : A \oplus B$ into $\Delta|\Gamma, z : A \oplus B, x : A \vdash_{\beta[\frac{x}{z}]} e_1 : C$ to get $\Delta|\Gamma, x : A \vdash_{(\beta[\frac{x}{z}])[\frac{\alpha}{z}]} e_1[\frac{inl(e')}{z}] : C$ We then weaken and cut $\Delta|\Gamma \vdash_\alpha e' : A$ into the previous result to get to get $\Delta|\Gamma \vdash_{((\beta[\frac{x}{z}])[\frac{\alpha}{z}])[\frac{\alpha}{x}]} (e_1[\frac{inl(e')}{z}])[\frac{e'}{x}] : C$. Now since $((\beta[\frac{x}{z}])[\frac{\alpha}{z}])[\frac{\alpha}{x}] \cong (\beta[\frac{x}{z}])[\frac{\alpha}{x}] \cong \beta[\frac{\alpha}{z}]$, we get

$$\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{z}]} (e_1[\frac{inl(e')}{z}])[\frac{e'}{x}] : C$$

$\oplus_{\boldsymbol{R_2}}$ and $\oplus_{\boldsymbol{L}}$

$$\oplus_{R_2}\frac{\dfrac{\Delta|\Gamma \vdash_\alpha e' : B}{\Delta|\Gamma \vdash_\alpha inr(e') : A \oplus B} \quad \dfrac{\Delta|\Gamma, z : A \oplus B, x : A \vdash_{\beta[\frac{x}{z}]} e_1 : C \qquad \Delta|\Gamma, z : A \oplus B, y : B \vdash_{\beta[\frac{y}{z}]} e_2 : C}{\Delta|\Gamma, z : A \oplus B \vdash_\beta \text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2 : C}\oplus_L}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{z}]} (\text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2)[\frac{inr(e')}{z}] : C}\,cut$$

Now we can weaken and cut $\Delta|\Gamma \vdash_\alpha inr(e') : A \oplus B$ into $\Delta|\Gamma, z : A \oplus B, y : B \vdash_{\beta[\frac{y}{z}]} e_2 : C$ to get $\Delta|\Gamma, y : B \vdash_{(\beta[\frac{y}{z}])[\frac{\alpha}{z}]} e_2[\frac{inr(e')}{z}] : C$ Then we can weaken and cut $\Delta|\Gamma \vdash_\alpha e' : B$ into the previous result to get $\Delta|\Gamma \vdash_{((\beta[\frac{y}{z}])[\frac{\alpha}{z}])[\frac{\alpha}{y}]} (e_2[\frac{inr(e')}{z}])[\frac{e'}{y}] : C$. Now since $((\beta[\frac{y}{z}])[\frac{\alpha}{z}])[\frac{\alpha}{y}] \cong (\beta[\frac{y}{z}])[\frac{\alpha}{y}] \cong \beta[\frac{\alpha}{z}]$, we get

$$\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{z}]} (e_2[\frac{inr(e')}{z}])[\frac{e'}{y}] : C$$

$\&_{\boldsymbol{R}}$ and $\&_{\boldsymbol{L_1}}$

$$\&_R\frac{\dfrac{\Delta|\Gamma \vdash_\alpha e_1 : A \qquad \Delta|\Gamma \vdash_\alpha e_2 : B}{\Delta|\Gamma \vdash_\alpha < e_1, e_2 > : A \,\&\, B} \quad \dfrac{\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\beta'} e : C \qquad \beta \cong \beta'[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_\beta \text{let } < x, \_ > \text{ be } z \text{ in } e : C}\&_{L_1}}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{z}]} (\text{let } < x, \_ > \text{ be } z \text{ in } e)[\frac{<e_1,e_2>}{z}] : C}\,cut$$

Now we can weaken and cut $\Delta|\Gamma \vdash_\alpha < e_1, e_2 > : A \,\&\, B$ into $\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\beta'} e : C$ to get $\Delta|\Gamma, x : A \vdash_{\beta'[\frac{\alpha}{z}]} e[\frac{<e_1,e_2>}{z}] : C$. Then we weaken and cut

$\Delta|\Gamma \vdash_\alpha e_1 : A$ into the previous result to get $\Delta|\Gamma \vdash_{(\beta'[\frac{\alpha}{z}])[\frac{\alpha}{x}]} (e[\frac{<e_1,e_2>}{z}])[\frac{e_1}{x}] : C$. We have $\beta[\frac{\alpha}{z}] \cong (\beta'[\frac{z}{x}])[\frac{\alpha}{z}] \cong (\beta'[\frac{\alpha}{z}])[\frac{\alpha}{x}]$. So we get

$$\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{z}]} (e[\frac{<e_1,e_2>}{z}])[\frac{e_1}{x}] : C$$

$\&_R$ and $\&_{L_2}$

$$\&_R \frac{\Delta|\Gamma \vdash_\alpha e_1 : A \qquad \Delta|\Gamma \vdash_\alpha e_2 : B}{\Delta|\Gamma \vdash_\alpha < e_1, e_2 > : A \,\&\, B} \quad \frac{\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\beta'} e : C \qquad \beta \cong \beta'[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_\beta \text{let } < \_, x > \text{ be } z \text{ in } e : C} \,\&_{L_2}$$
$$\frac{}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{z}]} (\text{let } < \_, x > \text{ be } z \text{ in } e)[\frac{<e_1,e_2>}{z}] : C} \, cut$$

Now we can weaken and cut $\Delta|\Gamma \vdash_\alpha < e_1, e_2 > : A \,\&\, B$ into $\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\beta'} e : C$ to get $\Delta|\Gamma, x : B \vdash_{\beta'[\frac{\alpha}{z}]} e[\frac{<e_1,e_2>}{z}] : C$. Then we weaken and cut $\Delta|\Gamma \vdash_\alpha e_2 : B$ into the previous result to get $\Delta|\Gamma \vdash_{(\beta'[\frac{\alpha}{z}])[\frac{\alpha}{x}]} (e[\frac{<e_1,e_2>}{z}])[\frac{e_2}{x}] : C$. We have $\beta[\frac{\alpha}{z}] \cong (\beta'[\frac{z}{x}])[\frac{\alpha}{z}] \cong (\beta'[\frac{\alpha}{z}])[\frac{\alpha}{x}]$.
So we get

$$\Delta|\Gamma \vdash_{\beta[\frac{\alpha}{z}]} (e[\frac{<e_1,e_2>}{z}])[\frac{e_2}{x}] : C$$

Now we consider the cases where the last step of the derivation of l is a left rule.

$\mathbf{1}_L$

$$1_L \frac{\dfrac{\Delta|\Gamma, z : 1 \vdash_{\alpha[\frac{\cdot}{z}]} e' : B}{\Delta|\Gamma, z : 1 \vdash_\alpha \text{let } * \text{ be } z \text{ in } e' : B} \qquad \Delta|\Gamma, z : 1, x : B \vdash_\beta e : A}{\Delta|\Gamma, z : 1 \vdash_{\beta[\frac{\alpha}{x}]} e[\frac{\text{let } * \text{ be } z \text{ in } e'}{x}] : A} \, cut$$

Now since we have $z : 1$ in the derivation of $\Delta|\Gamma, z : 1, x : B \vdash_\beta e : A$, by the inversion lemma we get $\Delta|\Gamma, z : 1, x : B \vdash_{\beta[\frac{\cdot}{z}]} e'' : A$. Now cut $\Delta|\Gamma, z : 1 \vdash_{\alpha[\frac{\cdot}{z}]} e' : B$ into $\Delta|\Gamma, z : 1, x : B \vdash_{\beta[\frac{\cdot}{z}]} e'' : A$ to get $\Delta|\Gamma, z : 1 \vdash_{(\beta[\frac{\cdot}{z}])[\frac{\alpha[\frac{\cdot}{z}]}{x}]} e''[\frac{e'}{x}] : A$. Now $(\beta[\frac{\cdot}{z}])[\frac{\alpha[\frac{\cdot}{z}]}{x}] \cong (\beta[\frac{\alpha}{x}])[\frac{\cdot}{z}]$, so we can apply 1_L.

$$1_L \frac{\Delta|\Gamma, z : 1 \vdash_{(\beta[\frac{\alpha}{x}])[\frac{\cdot}{z}]} e''[\frac{e'}{x}] : A}{\Delta|\Gamma, z : 1 \vdash_{\beta[\frac{\alpha}{x}]} \text{let } * \text{ be } z \text{ in } e''[\frac{e'}{x}] : B}$$

$\otimes_L$

$$\otimes_L \frac{\dfrac{\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e' : C}{\Delta|\Gamma, x : A \otimes B \vdash_\alpha \text{let } (x_1, x_2) \text{ be } x \text{ in } e' : C} \qquad \Delta|\Gamma, x : A \otimes B, y : C \vdash_\beta e : D}{\Delta|\Gamma, x : A \otimes B \vdash_{\beta[\frac{\alpha}{y}]} e[\frac{\text{let } (x_1, x_2) \text{ be } x \text{ in } e'}{y}] : D} \; cut$$

Now since we have $x : A \otimes B$ in the derivation of $\Delta|\Gamma, x : A \otimes B, y : C \vdash_\beta e : D$, so by the inversion lemma we get $\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B, y : C \vdash_{\beta[\frac{x_1 \cup x_2}{x}]} e'' : D$. Now cut $\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e' : C$ into $\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B, y : C \vdash_{\beta[\frac{x_1 \cup x_2}{x}]} e'' : D$ to get $\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{(\beta[\frac{x_1 \cup x_2}{x}])[\frac{\alpha[\frac{x_1 \cup x_2}{x}]}{y}]} e''[\frac{e'}{y}] : D$. Now $(\beta[\frac{x_1 \cup x_2}{x}])[\frac{\alpha[\frac{x_1 \cup x_2}{x}]}{y}] \cong (\beta[\frac{\alpha}{y}])[\frac{x_1 \cup x_2}{x}]$, so we can apply $\otimes_L$.

$$\otimes_L \frac{\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{(\beta[\frac{\alpha}{y}])[\frac{x_1 \cup x_2}{x}]} e''[\frac{e'}{y}] : D}{\Delta|\Gamma, x : A \otimes B \vdash_{\beta[\frac{\alpha}{y}]} \text{let } (x_1, x_2) \text{ be } x \text{ in } e''[\frac{e'}{y}] : C}$$

$\multimap_L$

$$\dfrac{\dfrac{\Delta|\Gamma, f : A \multimap B \vdash_{\alpha_1} t : A \qquad \Delta|\Gamma, f : A \multimap B, x : B \vdash_{\alpha_2} e' : C \qquad \alpha \cong \alpha_2[\frac{f \cup \alpha_1}{x}]}{\Delta|\Gamma, f : A \multimap B \vdash_{\alpha} \text{let } x \text{ be } f(t) \text{ in } e' : C} \multimap_L \qquad \Delta|\Gamma, f : A \multimap B, y : C \vdash_{\beta} e : D}{\Delta|\Gamma, f : A \multimap B \vdash_{\beta[\frac{\alpha}{y}]} e[\frac{\text{let } x \text{ be } f(t) \text{ in } e'}{y}] : D} \, cut$$

We cut $\Delta|\Gamma, f : A \multimap B, x : B \vdash_{\alpha_2} e' : C$ into a weakened $\Delta|\Gamma, f : A \multimap B, y : C \vdash_{\beta} e : D$ and we get $\Delta|\Gamma, f : A \multimap B, x : B \vdash_{\beta[\frac{\alpha_2}{y}]} e[\frac{e'}{y}] : D$. Now we have $(\beta[\frac{\alpha_2}{y}])[\frac{f \cup \alpha_1}{x}] \cong (\beta[\frac{f \cup \alpha_1}{x}])[\frac{\alpha_2[\frac{f \cup \alpha_1}{x}]}{y}] \cong \beta[\frac{\alpha}{y}]$ since $x \notin \beta$ and $\alpha \cong \alpha_2[\frac{f \cup \alpha_1}{x}]$. So finally we can apply $\multimap_L$ to get

$$\dfrac{\Delta|\Gamma, f : A \multimap B \vdash_{\alpha_1} t : A \qquad \Delta|\Gamma, f : A \multimap B, x : B \vdash_{\beta[\frac{\alpha_2}{y}]} e[\frac{e'}{y}] : D \qquad \beta[\frac{\alpha}{y}] \cong (\beta[\frac{\alpha_2}{y}])[\frac{f \cup \alpha_1}{x}]}{\Delta|\Gamma, f : A \multimap B \vdash_{\beta[\frac{\alpha}{y}]} \text{let } x \text{ be } f(t) \text{ in } e[\frac{e'}{y}] : C} \multimap_L$$

$\oplus_{\boldsymbol{L}}$

$$\dfrac{\dfrac{\Delta|\Gamma, z : A \oplus B, x : A \vdash_{\alpha[\frac{x}{z}]} e_1 : C \qquad \Delta|\Gamma, z : A \oplus B, y : B \vdash_{\alpha[\frac{y}{z}]} e_2 : C}{\Delta|\Gamma, z : A \oplus B \vdash_{\alpha} \text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2 : C} \oplus_L \qquad \Delta|\Gamma, z : A \oplus B, w : C \vdash_{\beta} e : D}{\Delta|\Gamma, z : A \oplus B \vdash_{\beta[\frac{\alpha}{w}]} e[\frac{\text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2}{w}] : D} \, cut$$

Now since we have $z : A \oplus B$ in the derivation of $\Delta|\Gamma, z : A \oplus B, w : C \vdash_{\beta} e : D$, by the inversion lemma e has a case in its body and can be rewritten as $e \cong \text{case } z \text{ of } inl(x) => e'_1, inr(y) => e'_2$. Hence we know $\Delta|\Gamma, z : A \oplus B, x : A, w : C \vdash_{\beta[\frac{x}{z}]} e'_1 : D$, and $\Delta|\Gamma, z : A \oplus B, y : B, w : C \vdash_{\beta[\frac{y}{z}]} e'_2 : D$. Now cut $\Delta|\Gamma, z : A \oplus B, x : A \vdash_{\alpha[\frac{x}{z}]} e_1 : C$ into $\Delta|\Gamma, z : A \oplus B, x : A, w : C \vdash_{\beta[\frac{x}{z}]} e'_1 : D$ to get $\Delta|\Gamma, z : A \oplus B, x : A \vdash_{(\beta[\frac{x}{z}])[\frac{\alpha[\frac{x}{z}]}{w}]} e'_1[\frac{e_1}{w}] : D$. Furthermore cut $\Delta|\Gamma, z : A \oplus B, y : B \vdash_{\alpha[\frac{y}{z}]} e_2 : C$ into $\Delta|\Gamma, z : A \oplus B, y : B, w : C \vdash_{\beta[\frac{y}{z}]} e'_2 : D$ to get $\Delta|\Gamma, z : A \oplus B, y : B \vdash_{(\beta[\frac{y}{z}])[\frac{\alpha[\frac{y}{z}]}{w}]} e'_2[\frac{e_2}{w}] : D$.

Now $(\beta[\frac{x}{z}])[\frac{\alpha[\frac{x}{z}]}{w}] \cong (\beta[\frac{\alpha}{w}])[\frac{x}{z}]$, and $(\beta[\frac{y}{z}])[\frac{\alpha[\frac{y}{z}]}{w}] \cong (\beta[\frac{\alpha}{w}])[\frac{y}{z}]$, so we can apply $\oplus_L$.

$$\frac{\Delta|\Gamma, z : A \oplus B, x : A \vdash_{(\beta[\frac{\alpha}{w}])[\frac{x}{z}]} e_1'[\frac{e_1}{w}] : D \qquad \Delta|\Gamma, z : A \oplus B, y : B \vdash_{(\beta[\frac{\alpha}{w}])[\frac{y}{z}]} e_2'[\frac{e_2}{w}] : D}{\Delta|\Gamma, z : A \oplus B \vdash_{\beta[\frac{\alpha}{w}]} \text{case } z \text{ of } inl(x) => e_1'[\frac{e_1}{w}], inr(y) => e_2'[\frac{e_2}{w}] : D} \oplus_L$$

$\&_{L_1}$

$$\frac{\dfrac{\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\alpha'} e' : C \qquad \alpha \cong \alpha'[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\alpha} \text{let } < x, \_ > \text{ be } z \text{ in } e' : C} \,\&_{L_1} \qquad \Delta|\Gamma, z : A \,\&\, B, w : C \vdash_{\beta} e : D}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\beta[\frac{\alpha}{w}]} e[\frac{\text{let } <x,\_> \text{ be } z \text{ in } e'}{w}] : D} \, cut$$

We cut $\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\alpha'} e' : C$ into a weakened $\Delta|\Gamma, z : A \,\&\, B, w : C \vdash_{\beta} e : D$ and we get $\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\beta[\frac{\alpha'}{w}]} e[\frac{e'}{w}] : D$. Now we have $(\beta[\frac{\alpha'}{w}])[\frac{z}{x}] \cong (\beta[\frac{z}{x}])[\frac{\alpha'[\frac{z}{x}]}{w}] \cong \beta[\frac{\alpha}{w}]$ since $x \notin \beta$ and $\alpha \cong \alpha'[\frac{z}{x}]$. So finally we can apply $\&_{L_1}$ to get

$$\frac{\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\beta[\frac{\alpha'}{w}]} e[\frac{e'}{w}] : D \qquad \beta[\frac{\alpha}{w}] \cong (\beta[\frac{\alpha'}{w}])[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\beta[\frac{\alpha}{w}]} \text{let } < x, \_ > \text{ be } z \text{ in } e[\frac{e'}{w}] : C} \,\&_{L_1}$$

$\&_{L_2}$

$$\frac{\dfrac{\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\alpha'} e' : C \qquad \alpha \cong \alpha'[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\alpha} \text{let } < \_, x > \text{ be } z \text{ in } e' : C} \,\&_{L_2} \qquad \Delta|\Gamma, z : A \,\&\, B, w : C \vdash_{\beta} e : D}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\beta[\frac{\alpha}{w}]} e[\frac{\text{let } <\_,x> \text{ be } z \text{ in } e}{w}] : D} \, cut$$

We cut $\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\alpha'} e' : C$ into a weakened $\Delta|\Gamma, z : A \,\&\, B, w : C \vdash_{\beta} e : D$ and we get $\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\beta[\frac{\alpha'}{w}]} e[\frac{e'}{w}] : D$. Now we have $(\beta[\frac{\alpha'}{w}])[\frac{z}{x}] \cong (\beta[\frac{z}{x}])[\frac{\alpha'[\frac{z}{x}]}{w}] \cong \beta[\frac{\alpha}{w}]$ since $x \notin \beta$ and $\alpha \cong \alpha'[\frac{z}{x}]$. So finally we can apply $\&_{L_2}$ to get

$$\frac{\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\beta[\frac{\alpha'}{w}]} e[\frac{e'}{w}] : D \qquad \beta[\frac{\alpha}{w}] \cong (\beta[\frac{\alpha'}{w}])[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\beta[\frac{\alpha}{w}]} \text{let } < \_, x > \text{ be } z \text{ in } e[\frac{e'}{w}] : C}\ \&_{L_1}$$

MV

$$\frac{\dfrac{\Delta, u|\Gamma \vdash_\gamma \theta : \Gamma_0 \qquad \Delta, u|\Gamma, z : A \vdash_\delta e' : C \qquad \delta[\frac{\alpha_0[\gamma]}{z}] \cong \alpha}{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_\alpha \text{let } z \text{ be } u[\theta] \text{ in } e' : C}\ MV \qquad \Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, x : C \vdash_\beta e : D}{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} e[\frac{\text{let } z \text{ be } u[\theta] \text{ in } e'}{x}] : D}\ cut$$

We cut $\Delta, u|\Gamma, z : A \vdash_\delta e' : C$ into a weakened $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, x : C \vdash_\beta e : D$ and we get $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_{\beta[\frac{\delta}{x}]} e[\frac{e'}{x}] : D$. Now we have $(\beta[\frac{\delta}{x}])[\frac{\alpha_0[\gamma]}{z}] \cong \beta[\frac{\delta[\frac{\alpha_0[\gamma]}{z}]}{x}] \cong \beta[\frac{\alpha}{x}]$ since $z \notin \beta$ and $\delta[\frac{\alpha_0[\gamma]}{z}] \cong \alpha$. So finally we can apply MV to get

$$\frac{\Delta, u|\Gamma \vdash_\gamma \theta : \Gamma_0 \qquad \Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_{\beta[\frac{\delta}{x}]} e[\frac{e'}{x}] : D \qquad (\beta[\frac{\delta}{x}])[\frac{\alpha_0[\gamma]}{z}] \cong \beta[\frac{\alpha}{x}]}{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_{\beta[\frac{\alpha}{x}]} \text{let } z \text{ be } u[\theta] \text{ in } e[\frac{e'}{x}] : C}\ MV$$

Hence we have completed our proof.

2. We write r for the derivation of A and s for the derivation of B. We proceed by induction on s.

Base cases:

$\mathbf{1_R}$

$$\frac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : A \qquad \dfrac{}{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_. * : 1}\ 1_R}{\Delta|\Gamma \vdash_. *[\frac{e'}{u}] : 1}\ cut!$$

This is trivial snce

$$\frac{}{\Delta|\Gamma \vdash_{\cdot} * : 1}\, 1_R$$

so we are done.

id

$$\frac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : B \qquad \dfrac{x : A \in \Gamma}{\Delta, u : [\Gamma_0]_{\alpha_0} B|\Gamma \vdash_x x : A}\,\mathrm{id}}{\Delta|\Gamma \vdash_x x[\frac{e'}{u}] : A}\, cut!$$

This is again trivial since

$$\frac{x : A \in \Gamma}{\Delta|\Gamma \vdash_x x : A}\,\mathrm{id}$$

so we are done.

Inductive cases:

$\mathbf{1_L}$

$$\frac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : B \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} B|\Gamma, z : 1 \vdash_{\alpha[\frac{1}{z}]} t : A}{\Delta, u : [\Gamma_0]_{\alpha_0} B|\Gamma, z : 1 \vdash_\alpha \text{let } 1 \text{ be } z \text{ in } t : A}\, 1_L}{\Delta|\Gamma \vdash_\alpha (\text{let } 1 \text{ be } z \text{ in } t)[\frac{e'}{u}] : A}\, cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : B$ and $\Delta, u : [\Gamma_0]_{\alpha_0} B|\Gamma, z : 1 \vdash_{\alpha[\frac{1}{z}]} t : A$ to get $\Delta|\Gamma, z : 1 \vdash_{\alpha[\frac{1}{z}]} t[\frac{e'}{u}] : A$. Then by applying 1_L we get

$$\frac{\Delta|\Gamma, z : 1 \vdash_{\alpha[\frac{1}{z}]} t[\frac{e'}{u}] : A}{\Delta|\Gamma, z : 1 \vdash_\alpha \text{let } 1 \text{ be } z \text{ in } t[\frac{e'}{u}] : A}\, 1_L$$

$\otimes_{\boldsymbol{R}}$

$$\dfrac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_{\alpha_1} e_1 : A \qquad \Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_{\alpha_2} e_2 : B \qquad \alpha \cong \alpha_1 \cup \alpha_2}{\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_{\alpha} (e_1, e_2) : A \otimes B} \otimes_R}{\Delta|\Gamma \vdash_{\alpha} (e_1, e_2)[\frac{e'}{u}] : A \otimes B}\, cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C$ and $\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_{\alpha_1} e_1 : A$ to get $\Delta|\Gamma \vdash_{\alpha_1} e_1[\frac{e'}{u}] : A$, as well as $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C$ and $\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_{\alpha_2} e_2 : B$. to get $\Delta|\Gamma \vdash_{\alpha_2} e_2[\frac{e'}{u}] : B$. Then by applying $\otimes_R$ we get

$$\dfrac{\Delta|\Gamma \vdash_{\alpha_1} e_1[\frac{e'}{u}] : A \qquad \Delta|\Gamma \vdash_{\alpha_2} e_2[\frac{e'}{u}] : B \qquad \alpha \cong \alpha_1 \cup \alpha_2}{\Delta|\Gamma \vdash_{\alpha} (e_1[\frac{e'}{u}], e_2[\frac{e'}{u}]) : A \otimes B} \otimes_R$$

$\otimes_{\boldsymbol{L}}$

$$\dfrac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e : C}{\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, x : A \otimes B \vdash_{\alpha} \text{let } (x_1, x_2) \text{ be } x \text{ in } e : C} \otimes_L}{\Delta|\Gamma, x : A \otimes B \vdash_{\alpha} (\text{let } (x_1, x_2) \text{ be } x \text{ in } e)[\frac{e'}{u}] : C}\, cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D$ and $\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e : C$ to get $\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e[\frac{e'}{u}] : C$. Then by applying $\otimes_L$ we get

$$\dfrac{\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e[\frac{e'}{u}] : C}{\Delta|\Gamma, x : A \otimes B \vdash_{\alpha} \text{let } (x_1, x_2) \text{ be } x \text{ in } e[\frac{e'}{u}] : C} \otimes_L$$

$\multimap_{\boldsymbol{R}}$

$$\frac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma, x : A \vdash_{\alpha \cup x} t : B}{\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_{\alpha} \lambda x.t : A \multimap B} \multimap_R}{\Delta|\Gamma \vdash_{\alpha} (\lambda x.t)[\frac{e'}{u}] : A \multimap B} \, cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C$ and $\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma, x : A \vdash_{\alpha \cup x} t : B$ to get $\Delta|\Gamma, x : A \vdash_{\alpha \cup x} t[\frac{e'}{u}] : B$. Then by applying $\multimap_R$ we get

$$\frac{\Delta|\Gamma, x : A \vdash_{\alpha \cup x} t[\frac{e'}{u}] : B}{\Delta|\Gamma \vdash_{\alpha} \lambda x.(t[\frac{e'}{u}]) : A \multimap B} \multimap_R$$

$\multimap_{\boldsymbol{L}}$

$$\frac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, f : A \multimap B \vdash_{\alpha_1} t : A \qquad \Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, f : A \multimap B, x : B \vdash_{\alpha_2} e : C \qquad \alpha \cong \alpha_2[\frac{f \cup \alpha_1}{x}]}{\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, f : A \multimap B \vdash_{\alpha} \text{let } x \text{ be } f(t) \text{ in } e : C} \multimap_L}{\Delta|\Gamma, f : A \multimap B \vdash_{\alpha} (\text{let } x \text{ be } f(t) \text{ in } e)[\frac{e'}{u}] : C} \, cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D$ and $\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, f : A \multimap B \vdash_{\alpha_1} t : A$ to get $\Delta|\Gamma, f : A \multimap B \vdash_{\alpha_1} t[\frac{e'}{u}] : A$, as well as $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D$ and $\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, f : A \multimap B, x : B \vdash_{\alpha_2} e : C$ to get $\Delta|\Gamma, f : A \multimap B, x : B \vdash_{\alpha_2} e[\frac{e'}{u}] : C$. Then by applying $\multimap_L$ we get

$$\frac{\Delta|\Gamma, f : A \multimap B \vdash_{\alpha_1} t[\frac{e'}{u}] : A \qquad \Delta|\Gamma, f : A \multimap B, x : B \vdash_{\alpha_2} e[\frac{e'}{u}] : C \qquad \alpha \cong \alpha_2[\frac{f \cup \alpha_1}{x}]}{\Delta|\Gamma, f : A \multimap B \vdash_{\alpha} \text{let } x \text{ be } f(t[\frac{e'}{u}]) \text{ in } e[\frac{e'}{u}] : C} \multimap_L$$

$\oplus_{\boldsymbol{R_1}}$

$$\frac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_{\alpha} t : A}{\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_{\alpha} inl(t) : A \oplus B} \oplus_{R_1}}{\Delta|\Gamma \vdash_{\alpha} inl(t)[\frac{e'}{u}] : A \oplus B} \, cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C$ and $\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_\alpha t : A$ to get $\Delta|\Gamma \vdash_\alpha t[\frac{e'}{u}] : A$. Then by applying $\oplus_{R_1}$ we get

$$\frac{\Delta|\Gamma \vdash_\alpha t[\frac{e'}{u}] : A}{\Delta|\Gamma \vdash_\alpha inl(t[\frac{e'}{u}]) : A \oplus B} \oplus_{R_1}$$

$\oplus_{\boldsymbol{R_2}}$

$$\frac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_\alpha t : B}{\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_\alpha inr(t) : A \oplus B} \oplus_{R_2}}{\Delta|\Gamma \vdash_\alpha inr(t)[\frac{e'}{u}] : A \oplus B} \, cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C$ and $\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_\alpha t : B$ to get $\Delta|\Gamma \vdash_\alpha t[\frac{e'}{u}] : B$. Then by applying $\oplus_{R_2}$ we get

$$\frac{\Delta|\Gamma \vdash_\alpha t[\frac{e'}{u}] : B}{\Delta|\Gamma \vdash_\alpha inr(t[\frac{e'}{u}]) : A \oplus B} \oplus_{R_2}$$

$\oplus_{\boldsymbol{L}}$

$$\frac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \oplus B, x : A \vdash_{\alpha[\frac{x}{z}]} e_1 : C \qquad \Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \oplus B, y : B \vdash_{\alpha[\frac{y}{z}]} e_2 : C}{\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \oplus B \vdash_\alpha \text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2 : C} \oplus_L}{\Delta|\Gamma, z : A \oplus B \vdash_\alpha (\text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2)[\frac{e'}{u}] : C} \, cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D$ and $\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \oplus B, x : A \vdash_{\alpha[\frac{x}{z}]} e_1 : C$ to get $\Delta|\Gamma, z : A \oplus B, x : A \vdash_{\alpha[\frac{x}{z}]} e_1[\frac{e'}{u}] : C$, as well as $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D$ and $\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \oplus B, y : B \vdash_{\alpha[\frac{y}{z}]} e_2 : C$ to get $\Delta|\Gamma, z : A \oplus B, y : B \vdash_{\alpha[\frac{y}{z}]} e_2[\frac{e'}{u}] : C$. Then by applying $\oplus_L$ we get

$$\frac{\Delta|\Gamma, z : A \oplus B, x : A \vdash_{\alpha[\frac{x}{z}]} e_1[\frac{e'}{u}] : C \qquad \Delta|\Gamma, z : A \oplus B, y : B \vdash_{\alpha[\frac{y}{z}]} e_2[\frac{e'}{u}] : C}{\Delta|\Gamma, z : A \oplus B \vdash_\alpha \text{case } z \text{ of } inl(x) => e_1[\frac{e'}{u}], inr(y) => e_2[\frac{e'}{u}] : C} \oplus_L$$

$\&_R$

$$\frac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_\alpha e_1 : A \qquad \Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_\alpha e_2 : B}{\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_\alpha < e_1, e_2 >: A \,\&\, B} \,\&_R}{\Delta|\Gamma \vdash_\alpha < e_1, e_2 > [\frac{e'}{u}] : A \,\&\, B} \, cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C$ and $\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_\alpha e_1 : A$ to get $\Delta|\Gamma \vdash_\alpha e_1[\frac{e'}{u}] : A$, as well as $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : C$ and $\Delta, u : [\Gamma_0]_{\alpha_0} C|\Gamma \vdash_\alpha e_2 : B$ to get $\Delta|\Gamma \vdash_\alpha e_2[\frac{e'}{u}] : B$. Then by applying $\&_R$ we get

$$\frac{\Delta|\Gamma \vdash_\alpha e_1[\frac{e'}{u}] : A \qquad \Delta|\Gamma \vdash_\alpha e_2[\frac{e'}{u}] : B}{\Delta|\Gamma \vdash_\alpha < e_1[\frac{e'}{u}], e_2[\frac{e'}{u}] >: A \,\&\, B} \,\&_R$$

$\&_{L_1}$

$$\frac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \,\&\, B, x : A \vdash_{\alpha'} e : C \qquad \alpha \cong \alpha'[\frac{z}{x}]}{\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \,\&\, B \vdash_\alpha \text{let } < x, \_ > \text{ be } z \text{ in } e : C} \,\&_{L_1}}{\Delta|\Gamma, z : A \,\&\, B \vdash_\alpha (\text{let } < x, \_ > \text{ be } z \text{ in } e)[\frac{e'}{u}] : C} \, cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D$ and $\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \,\&\, B, x : A \vdash_{\alpha'} e : C$ to get $\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\alpha'} e[\frac{e'}{u}] : C$. Then by applying $\&_{L_1}$ we get

$$\frac{\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\alpha'} e[\frac{e'}{u}] : C \qquad \alpha \cong \alpha'[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_\alpha \text{let } < x, \_ > \text{ be } z \text{ in } e[\frac{e'}{u}] : C} \,\&_{L_1}$$

&L₂

$$\dfrac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D \qquad \dfrac{\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \,\&\, B, x : B \vdash_{\alpha'} e : C \qquad \alpha \cong \alpha'[\frac{z}{x}]}{\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \,\&\, B \vdash_{\alpha} \text{let } < \_, x > \text{ be } z \text{ in } e : C}\,\&_{L_2}}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\alpha} (\text{let } < \_, x > \text{ be } z \text{ in } e)[\frac{e'}{u}] : C}\,cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : D$ and $\Delta, u : [\Gamma_0]_{\alpha_0} D|\Gamma, z : A \,\&\, B, x : B \vdash_{\alpha'} e : C$ to get $\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\alpha'} e[\frac{e'}{u}] : C$. Then by applying $\&_{L_2}$ we get

$$\dfrac{\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\alpha'} e[\frac{e'}{u}] : C \qquad \alpha \cong \alpha'[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\alpha} \text{let } < \_, x > \text{ be } z \text{ in } e[\frac{e'}{u}] : C}\,\&_{L_2}$$

MV

We now have 2 different cases:

First we look at the case when u is the variable the MV rule is applied to:

$$\dfrac{\Delta|\Gamma_0 \vdash_{\alpha_0} e' : A \qquad \dfrac{\Delta, u|\Gamma \vdash_{\gamma} \theta : \Gamma_0 \qquad \Delta, u|\Gamma, z : A \vdash_{\beta} e : C \qquad \beta[\frac{\alpha_0[\gamma]}{z}] \cong \alpha}{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_{\alpha} \text{let } z \text{ be } u[\theta] \text{ in } e : C}\,MV}{\Delta|\Gamma \vdash_{\alpha} (\text{let } z \text{ be } u[\theta] \text{ in } e)[\frac{e'}{u}] : C}\,cut!$$

Now we can cut $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : A$ and $\Delta, u|\Gamma, z : A \vdash_{\beta} e : C$ to get $\Delta|\Gamma, z : A \vdash_{\beta} e[\frac{e'}{u}] : C$, as well as cutting $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : A$ into $\Delta, u|\Gamma \vdash_{\gamma} \theta : \Gamma_0$ to get $\Delta|\Gamma \vdash_{\gamma} \theta[\frac{e'}{u}] : \Gamma_0$. Now composing this with $\Delta|\Gamma_0 \vdash_{\alpha_0} e' : A$ we get $\Delta|\Gamma \vdash_{\alpha_0[\gamma]} e'[\theta[\frac{e'}{u}]] : A$. Then by applying *cut* we get

$$\dfrac{\Delta|\Gamma \vdash_{\alpha_0[\gamma]} e'[\theta[\frac{e'}{u}]] : A \qquad \Delta|\Gamma, z : A \vdash_{\beta} e[\frac{e'}{u}] : C}{\Delta|\Gamma \vdash_{\beta[\frac{\alpha_0[\gamma]}{z}]} (e[\frac{e'}{u}])[\frac{e'[\theta[\frac{e'}{u}]]}{z}] : C}\,cut$$

Now since $\beta[\frac{\alpha_0[\gamma]}{z}] \cong \alpha$, we are done.

The second case is when the MV rule is not applied to u:

$$\dfrac{\Delta, v : [\Gamma_0]_{\alpha_0} A | \Gamma_1 \vdash_{\alpha_1} e' : B \qquad \dfrac{\Delta, u, v | \Gamma \vdash_\gamma \theta : \Gamma_0 \qquad \Delta, u, v | \Gamma, z : A \vdash_\beta e : C \qquad \beta[\frac{\alpha_0[\gamma]}{z}] \cong \alpha}{\Delta, u : [\Gamma_1]_{\alpha_1} B, v : [\Gamma_0]_{\alpha_0} A | \Gamma \vdash_\alpha \text{let } z \text{ be } v[\theta] \text{ in } e : C}\, MV}{\Delta | \Gamma, v : [\Gamma_0]_{\alpha_0} A \vdash_\alpha (\text{let } z \text{ be } v[\theta] \text{ in } e)[\frac{e'}{u}] : C}\, cut!$$

Now we can cut $\Delta, v : [\Gamma_0]_{\alpha_0} A | \Gamma_1 \vdash_{\alpha_1} e' : B$ into $\Delta, u, v | \Gamma \vdash_\gamma \theta : \Gamma_0$ to get $\Delta, v | \Gamma \vdash_\gamma \theta[\frac{e'}{u}] : \Gamma_0$. Then we cut $\Delta, v : [\Gamma_0]_{\alpha_0} A | \Gamma_1 \vdash_{\alpha_1} e' : B$ into $\Delta, u, v | \Gamma, z : A \vdash_\beta e : C$ to get $\Delta, v | \Gamma, z : A \vdash_\beta e[\frac{e'}{u}] : C$. Now we reapply MV to get

$$\dfrac{\Delta, v | \Gamma \vdash_\gamma \theta[\frac{e'}{u}] : \Gamma_0 \qquad \Delta, v | \Gamma, z : A \vdash_\beta e[\frac{e'}{u}] : C \qquad \beta[\frac{\alpha_0[\gamma]}{z}] \cong \alpha}{\Delta, v : [\Gamma_0]_{\alpha_0} A | \Gamma \vdash_\alpha \text{let } z \text{ be } v(\theta[\frac{e'}{u}]) \text{ in } e[\frac{e'}{u}] : C}\, MV$$

□

4. An Implementation Sequent Calculus

In order to implement our proof assistant, we had to translate the base sequent calculus from Figure 13 to rules that we can easily implement. The main challenge is representing the αs and the equations they are abiding. We introduce resource variables $(a_1, a_2, ...)$ to stand for $(\alpha_1, \alpha_2, ...)$ and a restriction context $\mathcal{X}$ which keeps track of all of the restrictions that will help us determine appropriate αs. The rules of this new sequent calculus are mirroring the rules from the base sequent calculus, except that each sequent now also contains a $\mathcal{X}$ context.

The restrictions take one of the following 6 forms:

1. $a = \{x_1, x_2, ..., x_n\}$ which tells us that α must contain exactly the variables $\{x_1, x_2, ..., x_n\}$.
2. $a = a_1 \cup a_2$ which tells us that $\alpha \cong \alpha_1 \cup \alpha_2$.
3. $a[\frac{\vec{x}}{\vec{y}}] = a'$ which tells us that $\alpha[\frac{\vec{x}}{\vec{y}}] \cong \alpha'$.
4. $a = a'[\frac{\vec{x}}{\vec{y}}]$ which tells us that $\alpha \cong \alpha'[\frac{\vec{x}}{\vec{y}}]$.
5. $a = a_2[\frac{f \cup a_1}{x}]$ which tells us that $\alpha \cong \alpha_2[\frac{f \cup \alpha_1}{x}]$.
6. $a = a'[\frac{a_0[\gamma]}{z}]$ which tells us that $\alpha \cong \alpha'[\frac{\alpha_0[\gamma]}{z}]$.

We say that $\mathcal{X} \vDash \frac{\alpha_1}{a_1}, \frac{\alpha_2}{a_2}, ..., \frac{\alpha_n}{a_n}$, or $\mathcal{X}$ is consistent, if there are $\alpha_1, ..., \alpha_n$ for which the restrictions in $\mathcal{X}$ hold. Note that this does not uniquely identify the α's since in some cases (like splitting an α during a tensor right rule and then only having metavariables as terms on both sides), the α's have many different values for which all restrictions hold.

In order to be able to tell which rules can be applied to which variables under every a during the process of building the proof, we can compute the upper bound

multiset of variables each a can use by solving all of the restrictions in $\mathcal{X}$. We use the following notation: $a \subset_{\mathcal{X}} \{x_1, ..., x_n\}$ to say that by solving the restrictions in $\mathcal{X}$ we can generate the upper bound $x_1, ..., x_n$. In the implementation we cache these results and just update them by subtracting variables as they are being used in other branches of the proof. In Section 4.1 we prove a theorem that from a derivation of a term in the sequent calculus from Figure 14, we can produce a derivation of a term of the same type in the sequent calculus from Figure 13.

4.1. From an implementation calculus proof to a base calculus proof.

We want to prove that if we have a consistent set of constraints $\mathcal{X}$ and a proof of a theorem in our implementation calculus, then we can construct a proof of the same theorem in our base calculus as well.

Theorem:
If $\mathcal{X} \vDash \frac{\alpha_1}{a_1}, ..., \frac{\alpha_n}{a_n}$ and $\mathcal{X}|u_0 : [\Gamma_0]_{a_0} A_0, ..., u_k : [\Gamma_k]_{a_k} A_k|\Gamma \vdash_{a_i} A$,
then $u_0 : [\Gamma_0]_{\alpha_0} A_0, ..., u_k : [\Gamma_k]_{\alpha_k} A_k|\Gamma \vdash_{\alpha_i} A$.

Proof.
Base cases:

1_R

$$\frac{(a = \{\}) \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a * : 1}\, 1_R$$

Now since $(a = \{\}) \in \mathcal{X}$, and $\mathcal{X} \vDash \frac{\alpha}{a}$, $\alpha \cong \cdot$. Therefore,

$$\Gamma ::= \cdot \mid x : A \mid \Gamma_1 \cup \Gamma_2$$

$$\Delta ::= \cdot \mid u : [\Gamma_0]_{a_0} A \mid \Delta_1 \cup \Delta_2$$

$$\gamma ::= \cdot \mid \gamma,\ \tfrac{a_i}{y_i}$$

$$\mathcal{X} ::= \cdot \mid \mathcal{X},\ a = \{x_1, x_2, ..., x_n\} \mid \mathcal{X},\ a = a_1 \cup a_2 \mid \mathcal{X},\ a[\tfrac{\vec{x}}{\vec{y}}] = a' \mid \mathcal{X},\ a = a'[\tfrac{\vec{x}}{\vec{y}}]$$
$$\mid \mathcal{X},\ a = a_2[\tfrac{f \cup a_1}{x}] \mid \mathcal{X},\ a = a'[\tfrac{a_0[\gamma]}{z}]$$

$$\frac{(a = \{\}) \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a * : 1}\, 1_R \qquad \frac{\mathcal{X}|\Delta|\Gamma, z : 1 \vdash_{a'} t : A \qquad a[\tfrac{\{\}}{\{z\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, z : 1 \vdash_a \text{let } * \text{ be } z \text{ in } t : A}\, 1_L \qquad \frac{x : A \in \Gamma \qquad (a = \{x\}) \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a x : A}\, \text{id}$$

$$\frac{\mathcal{X}|\Delta|\Gamma \vdash_{a_1} e_1 : A \qquad \mathcal{X}|\Delta|\Gamma \vdash_{a_2} e_2 : B \qquad a = a_1 \cup a_2 \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a (e_1, e_2) : A \otimes B}\, \otimes_R$$

$$\frac{\mathcal{X}|\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{a'} e : C \qquad a[\tfrac{\{x_1, x_2\}}{\{x\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, x : A \otimes B \vdash_a \text{let } (x_1, x_2) \text{ be } x \text{ in } e : C}\, \otimes_L$$

$$\frac{\mathcal{X}|\Delta|\Gamma, x : A \vdash_{a'} t : B \qquad a[\tfrac{\{x\}}{\{\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a \lambda x.t : A \multimap B}\, \multimap_R$$

$$\frac{\mathcal{X}|\Delta|\Gamma, f : A \multimap B \vdash_{a_1} t : A \qquad \mathcal{X}|\Delta|\Gamma, f : A \multimap B, x : B \vdash_{a_2} e : C \qquad a = a_2[\tfrac{f \cup a_1}{x}] \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, f : A \multimap B \vdash_a \text{let } x \text{ be } f(t) \text{ in } e : C}\, \multimap_L$$

$$\frac{\mathcal{X}|\Delta|\Gamma \vdash_{a'} t : A \qquad a[\tfrac{\{\}}{\{\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a inl(t) : A \oplus B}\, \oplus_{R_1} \qquad \frac{\mathcal{X}|\Delta|\Gamma \vdash_{a'} t : B \qquad a[\tfrac{\{\}}{\{\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a inr(t) : A \oplus B}\, \oplus_{R_2}$$

$$\frac{\mathcal{X}|\Delta|\Gamma, z, x : A \vdash_{a_1} e_1 : C \qquad \mathcal{X}|\Delta|\Gamma, z, y : B \vdash_{a_2} e_2 : C \qquad a[\tfrac{\{x\}}{\{z\}}] = a_1 \in \mathcal{X} \qquad a[\tfrac{\{y\}}{\{z\}}] = a_2 \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, z : A \oplus B \vdash_a \text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2 : C}\, \oplus_L$$

$$\frac{\mathcal{X}|\Delta|\Gamma \vdash_{a_1} e_1 : A \qquad \mathcal{X}|\Delta|\Gamma \vdash_{a_2} e_2 : B \qquad a[\tfrac{\{\}}{\{\}}] = a_1 \in \mathcal{X} \qquad a[\tfrac{\{\}}{\{\}}] = a_2 \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a < e_1, e_2 > : A \,\&\, B}\, \&_R$$

$$\frac{\mathcal{X}|\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{a'} e : C \qquad a = a'[\tfrac{\{z\}}{\{x\}}] \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, z : A \,\&\, B \vdash_a \text{let } < x, \_ > \text{ be } z \text{ in } e : C}\, \&_{L_1}$$

$$\frac{\mathcal{X}|\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{a'} e : C \qquad a = a'[\tfrac{\{z\}}{\{x\}}] \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, z : A \,\&\, B \vdash_a \text{let } < \_, x > \text{ be } z \text{ in } e : C}\, \&_{L_2}$$

$$\frac{\mathcal{X}|\Delta, u|\Gamma \vdash_\gamma \theta : \Gamma_0 \qquad \mathcal{X}|\Delta, u|\Gamma, z : A \vdash_{a'} e : C \qquad a = a'[\tfrac{a_0[\gamma]}{z}] \in \mathcal{X}}{\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A|\Gamma \vdash_a \text{let } z \text{ be } u[\theta] \text{ in } e : C}\, MV$$

FIGURE 14. Implementation Sequent Calculus

$$\frac{}{\Delta|\Gamma \vdash_\alpha * : 1}\, 1_R$$

id

$$\frac{x : A \in \Gamma \qquad (a = \{x\}) \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a x : A}\,\text{id}$$

Now since $(a = \{x\}) \in \mathcal{X}$, and $\mathcal{X} \vDash \frac{\alpha}{a}$, $\alpha \cong x$. Therefore,

$$\frac{x : A \in \Gamma}{\Delta|\Gamma \vdash_\alpha x : A}\,\text{id}$$

Inductive cases:

1_L

$$\frac{\mathcal{X}|\Delta|\Gamma, z : 1 \vdash_{a'} t : A \qquad a[\frac{\{\}}{\{z\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, z : 1 \vdash_a \text{let } * \text{ be } z \text{ in } t : A}\, 1_L$$

Now by applying IH to $\mathcal{X}|\Delta|\Gamma, z : 1 \vdash_{a'} t : A$ we get $\Delta|\Gamma, z : 1 \vdash_{\alpha'} t : A$. Now since $a[\frac{\{\}}{\{z\}}] = a' \in \mathcal{X}$, and $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha'}{a'}, \alpha[\frac{\cdot}{z}] \cong \alpha'$, so by applying 1_L we get

$$\frac{\Delta|\Gamma, z : 1 \vdash_{\alpha[\frac{\cdot}{z}]} t : A}{\Delta|\Gamma, z : 1 \vdash_\alpha \text{let } * \text{ be } z \text{ in } t : A}\, 1_L$$

$\otimes_R$

$$\frac{\mathcal{X}|\Delta|\Gamma \vdash_{a_1} e_1 : A \qquad \mathcal{X}|\Delta|\Gamma \vdash_{a_2} e_2 : B \qquad a = a_1 \cup a_2 \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a (e_1, e_2) : A \otimes B}\, \otimes_R$$

Now by applying IH to $\mathcal{X}|\Delta|\Gamma \vdash_{a_1} e_1 : A$ and $\mathcal{X}|\Delta|\Gamma \vdash_{a_2} e_2 : B$ we get $\Delta|\Gamma \vdash_{\alpha_1} e_1 : A$ and $\Delta|\Gamma \vdash_{\alpha_2} e_2 : B$. Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha_1}{a_1}, \frac{\alpha_2}{a_2}$, and $a = a_1 \cup a_2 \in \mathcal{X}$, we have $\alpha \cong \alpha_1 \cup \alpha_2$. Then we can apply $\otimes_R$ again to get

$$\frac{\Delta|\Gamma \vdash_{\alpha_1} e_1 : A \qquad \Delta|\Gamma \vdash_{\alpha_2} e_2 : B \qquad \alpha \cong \alpha_1 \cup \alpha_2}{\Delta|\Gamma \vdash_{\alpha} (e_1, e_2) : A \otimes B} \otimes_R$$

$\boldsymbol{\otimes_L}$

$$\frac{\mathcal{X}|\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{a'} e : C \qquad a[\frac{\{x_1,x_2\}}{\{x\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, x : A \otimes B \vdash_{a} \text{let } (x_1, x_2) \text{ be } x \text{ in } e : C} \otimes_L$$

By applying the IH to $\mathcal{X}|\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{a'} e : C$ we get $\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\alpha'} e : C$. Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha'}{a'}$ and $a[\frac{\{x_1,x_2\}}{\{x\}}] = a' \in \mathcal{X}$, we have $\alpha[\frac{x_1 \cup x_2}{x}] \cong \alpha'$. Now we can reapply $\otimes_L$ to get

$$\frac{\Delta|\Gamma, x : A \otimes B, x_1 : A, x_2 : B \vdash_{\alpha[\frac{x_1 \cup x_2}{x}]} e : C}{\Delta|\Gamma, x : A \otimes B \vdash_{\alpha} \text{let } (x_1, x_2) \text{ be } x \text{ in } e : C} \otimes_L$$

$\boldsymbol{\multimap_R}$

$$\frac{\mathcal{X}|\Delta|\Gamma, x : A \vdash_{a'} t : B \qquad a[\frac{\{x\}}{\{\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_{a} \lambda x.t : A \multimap B} \multimap_R$$

By applying the IH to $\mathcal{X}|\Delta|\Gamma, x : A \vdash_{a'} t : B$ we get $\Delta|\Gamma, x : A \vdash_{\alpha'} t : B$. Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha'}{a'}$ and $a[\frac{\{x\}}{\{\}}] = a' \in \mathcal{X}$, we have $\alpha \cup x \cong \alpha'$. Now we can reapply $\multimap_R$ to get

$$\frac{\Delta|\Gamma, x : A \vdash_{\alpha \cup x} t : B}{\Delta|\Gamma \vdash_{\alpha} \lambda x.t : A \multimap B} \multimap_R$$

$\multimap_L$

$$\frac{\mathcal{X}|\Delta|\Gamma, f : A \multimap B \vdash_{a_1} t : A \qquad \mathcal{X}|\Delta|\Gamma, f : A \multimap B, x : B \vdash_{a_2} e : C \qquad a = a_2[\frac{f \cup a_1}{x}] \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, f : A \multimap B \vdash_a \text{let } x \text{ be } f(t) \text{ in } e : C} \multimap_L$$

By applying the IH to $\mathcal{X}|\Delta|\Gamma, f : A \multimap B \vdash_{a_1} t : A$ and $\mathcal{X}|\Delta|\Gamma, f : A \multimap B, x : B \vdash_{a_2} e : C$ we get $\Delta|\Gamma, f : A \multimap B \vdash_{\alpha_1} t : A$ and $\Delta|\Gamma, f : A \multimap B, x : B \vdash_{\alpha_2} e : C$. Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha_1}{a_1}, \frac{\alpha_2}{a_2}$, and $a = a_2[\frac{f \cup a_1}{x}] \in \mathcal{X}$, we have $\alpha \cong \alpha_2[\frac{f \cup \alpha_1}{x}]$. Then we can apply $\multimap_L$ again to get

$$\frac{\Delta|\Gamma, f : A \multimap B \vdash_{\alpha_1} t : A \qquad \Delta|\Gamma, f : A \multimap B, x : B \vdash_{\alpha_2} e : C \qquad \alpha \cong \alpha_2[\frac{f \cup \alpha_1}{x}]}{\Delta|\Gamma, f : A \multimap B \vdash_\alpha \text{let } x \text{ be } f(t) \text{ in } e : C} \multimap_L$$

$\oplus_{R_1}$

$$\frac{\mathcal{X}|\Delta|\Gamma \vdash_{a'} t : A \qquad a[\frac{\{\}}{\{\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a inl(t) : A \oplus B} \oplus_{R_1}$$

By applying IH to $\mathcal{X}|\Delta|\Gamma \vdash_{a'} t : A$ we get $\Delta|\Gamma \vdash_{\alpha'} t : A$. Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha'}{a'}$ and $a[\frac{\{\}}{\{\}}] = a' \in \mathcal{X}$, we have $\alpha \cong \alpha'$. Now we can reapply $\oplus_{R_1}$ to get

$$\frac{\Delta|\Gamma \vdash_\alpha t : A}{\Delta|\Gamma \vdash_\alpha inl(t) : A \oplus B} \oplus_{R_1}$$

$\oplus_{R_2}$

$$\frac{\mathcal{X}|\Delta|\Gamma \vdash_{a'} t : B \qquad a[\frac{\{\}}{\{\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_a inr(t) : A \oplus B} \oplus_{R_2}$$

By applying IH to $\mathcal{X}|\Delta|\Gamma \vdash_{a'} t : B$ we get $\Delta|\Gamma \vdash_{\alpha'} t : B$. Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha'}{a'}$ and $a[\frac{\{\}}{\{\}}] = a' \in \mathcal{X}$, we have $\alpha \cong \alpha'$. Now we can reapply $\oplus_{R_2}$ to get

$$\frac{\Delta|\Gamma \vdash_{\alpha} t : B}{\Delta|\Gamma \vdash_{\alpha} inr(t) : A \oplus B} \oplus_{R_2}$$

$\oplus_{\boldsymbol{L}}$

$$\frac{\mathcal{X}|\Delta|\Gamma, z, x : A \vdash_{a_1} e_1 : C \qquad \mathcal{X}|\Delta|\Gamma, z, y : B \vdash_{a_2} e_2 : C \qquad a[\frac{\{x\}}{\{z\}}] = a_1 \in \mathcal{X} \qquad a[\frac{\{y\}}{\{z\}}] = a_2 \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, z : A \oplus B \vdash_{a} \text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2 : C} \oplus_L$$

By applying IH to $\mathcal{X}|\Delta|\Gamma, z, x : A \vdash_{a_1} e_1 : C$ and $\mathcal{X}|\Delta|\Gamma, z, y : B \vdash_{a_2} e_2 : C$ we get $\Delta|\Gamma, z, x : A \vdash_{\alpha_1} e_1 : C$ and $\Delta|\Gamma, z, y : B \vdash_{\alpha_2} e_2 : C$. Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha_1}{a_1}, \frac{\alpha_2}{a_2}$, and $a[\frac{\{x\}}{\{z\}}] = a_1 \in \mathcal{X}$ and $a[\frac{\{y\}}{\{z\}}] = a_2 \in \mathcal{X}$, we get $\alpha[\frac{x}{z}] \cong \alpha_1$ and $\alpha[\frac{y}{z}] \cong \alpha_2$. Now we can reapply $\oplus_L$ to get

$$\frac{\Delta|\Gamma, z, x : A \vdash_{\alpha[\frac{x}{z}]} e_1 : C \qquad \Delta|\Gamma, z, y : B \vdash_{\alpha[\frac{y}{z}]} e_2 : C}{\Delta|\Gamma, z : A \oplus B \vdash_{\alpha} \text{case } z \text{ of } inl(x) => e_1, inr(y) => e_2 : C} \oplus_L$$

$\&_{\boldsymbol{R}}$

$$\frac{\mathcal{X}|\Delta|\Gamma \vdash_{a_1} e_1 : A \qquad \mathcal{X}|\Delta|\Gamma \vdash_{a_2} e_2 : B \qquad a[\frac{\{\}}{\{\}}] = a_1 \in \mathcal{X} \qquad a[\frac{\{\}}{\{\}}] = a_2 \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma \vdash_{a} < e_1, e_2 > : A \,\&\, B} \&_R$$

By applying IH to $\mathcal{X}|\Delta|\Gamma \vdash_{a_1} e_1 : A$ and $\mathcal{X}|\Delta|\Gamma \vdash_{a_2} e_2 : B$ we get $\Delta|\Gamma \vdash_{\alpha_1} e_1 : A$ and $\Delta|\Gamma \vdash_{\alpha_2} e_2 : B$. Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha_1}{a_1}, \frac{\alpha_2}{a_2}$, and $a[\frac{\{\}}{\{\}} = a_1] \in \mathcal{X}$ and $a[\frac{\{\}}{\{\}}] = a_2 \in \mathcal{X}$, we get $\alpha \cong \alpha_1$ and $\alpha \cong \alpha_2$. Now we can reapply $\&_R$ to get

$$\frac{\Delta|\Gamma \vdash_{\alpha_1} e_1 : A \qquad \Delta|\Gamma \vdash_{\alpha_2} e_2 : B}{\Delta|\Gamma \vdash_{\alpha} < e_1, e_2 > : A \,\&\, B} \&_R$$

$\&_{L_1}$

$$\frac{\mathcal{X}|\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{a'} e : C \qquad a[\frac{\{x\}}{\{z\}}] = a' \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, z : A \,\&\, B \vdash_a \text{let } < x, \_ > \text{ be } z \text{ in } e : C} \,\&_{L_1}$$

By applying IH to $\mathcal{X}|\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{a'} e : C$ we get $\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\alpha'} e : C$. Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha'}{a'}$ and $a = a'[\frac{\{z\}}{\{x\}}] \in \mathcal{X}$, we have $\alpha \cong \alpha'[\frac{z}{x}]$. Now we can reapply $\&_{L_2}$ to get

$$\frac{\Delta|\Gamma, z : A \,\&\, B, x : A \vdash_{\alpha'} e : C \qquad \alpha \cong \alpha'[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\alpha} \text{let } < x, \_ > \text{ be } z \text{ in } e : C} \,\&_{L_1}$$

$\&_{L_2}$

$$\frac{\mathcal{X}|\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{a'} e : C \qquad a = a'[\frac{\{z\}}{\{x\}}] \in \mathcal{X}}{\mathcal{X}|\Delta|\Gamma, z : A \,\&\, B \vdash_a \text{let } < \_, x > \text{ be } z \text{ in } e : C} \,\&_{L_2}$$

By applying IH to $\mathcal{X}|\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{a'} e : C$ we get $\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\alpha'} e : C$. Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha'}{a'}$ and $a = a'[\frac{\{z\}}{\{x\}}] \in \mathcal{X}$, we have $\alpha \cong \alpha'[\frac{z}{x}]$. Now we can reapply $\&_{L_2}$ to get

$$\frac{\Delta|\Gamma, z : A \,\&\, B, x : B \vdash_{\alpha'} e : C \qquad \alpha \cong \alpha'[\frac{z}{x}]}{\Delta|\Gamma, z : A \,\&\, B \vdash_{\alpha} \text{let } < \_, x > \text{ be } z \text{ in } e : C} \,\&_{L_2}$$

MV

$$\frac{\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A|\Gamma \vdash_\gamma \theta : \Gamma_0 \qquad \mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A|\Gamma, z : A \vdash_{a'} e : C \qquad a = a'[\frac{a_0[\gamma]}{z}] \in \mathcal{X}}{\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A|\Gamma \vdash_a \text{let } z \text{ be } u[\theta] \text{ in } e : C} \, MV$$

By applying IH to $\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A|\Gamma, z : A \vdash_{a'} e : C$ and $\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A|\Gamma \vdash_\gamma \theta : \Gamma_0$ we get $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, z : A \vdash_{\alpha'} e : C$ and $\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_\gamma \theta : \Gamma_0$ Now since $\mathcal{X} \vDash \frac{\alpha}{a}, \frac{\alpha'}{a'}, \frac{\alpha_0}{a_0}$ and $a = a'[\frac{a_0[\gamma]}{z}] \in \mathcal{X}$, we have $\alpha \cong \alpha'[\frac{\alpha_0[\gamma]}{z}]$. Now we can reapply MV to get

$$\frac{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_\gamma \theta : \Gamma_0 \qquad \Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma, z : A \vdash_{\alpha'} e : C \qquad \alpha \cong \alpha'[\frac{\alpha_0[\gamma]}{z}]}{\Delta, u : [\Gamma_0]_{\alpha_0} A|\Gamma \vdash_\alpha \text{let } z \text{ be } u[\theta] \text{ in } e : C}\, MV$$

□

4.2. Validity preservation.

We want to show that at every step of our term building process we have a term that typechecks and whose restriction context is consistent, so therefore by the theorem in Section 4.1, it has a proof in our base logical sequent calculus too.

We first show that we always start with a term that typechecks and has a consistent starting restriction context. Since we only construct a term made of a single metavariable whose context is the whole context entered so far, and whose restrictions are true for $\alpha \cong \Gamma$ and $\alpha' \cong z$, the proof is the following:

First let $M =$

$$\frac{z \in \Gamma, z : A \qquad (a' = \{z\}) \in a' = \{z\}, a = \Gamma, a = a'[\frac{a[id]}{z}]}{a' = \{z\}, a = \Gamma, a = a'[\frac{a[id]}{z}]|u : [\Gamma]_a A|\Gamma, z : A \vdash_{a'} z : A}\, id$$

So we have

$$\frac{\Gamma \vdash_{id} id : \Gamma \qquad M \qquad a = a'[\frac{a[id]}{z}] \in a' = \{z\}, a = \Gamma, a = a'[\frac{a[id]}{z}]}{a' = \{z\}, a = \Gamma, a = a'[\frac{a[id]}{z}]|u : [\Gamma]_a A|\Gamma \vdash_a \text{let } z \text{ be } u[id] \text{ in } z : A}\, MV$$

Lemma 1 (Δ weakening):

If $\mathcal{X}|\Delta|\Gamma \vdash_a A$ then $\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A_0|\Gamma \vdash_a A$, where a_0 is a fresh resource variable.

PROOF. By induction on $\mathcal{X}|\Delta|\Gamma \vdash_a A$. □

Lemma 2 ($\mathcal{X}$ weakening):

If $\mathcal{X}|\Delta|\Gamma \vdash_a A$ then $\mathcal{X}, \mathcal{X}'|\Delta|\Gamma \vdash_a A$.

PROOF. By induction on $\mathcal{X}|\Delta|\Gamma \vdash_a A$. □

Cut! rule for the $\mathcal{X}$ calculus:

If $\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A_0|\Gamma \vdash_a A$, and $\mathcal{X}|\Delta|\Gamma_0 \vdash_{a_0} A_0$, then $\mathcal{X}|\Delta|\Gamma \vdash_a A$.

PROOF. By induction that follows the same structure as the cut! proof in Section 3.3. □

Corollary:

If $\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A_0|\Gamma \vdash_a e : A$ and e contains let z be $u[\theta]$ in z in it, and we have $\mathcal{X}, \mathcal{X}'|\Delta, \Delta'|\Gamma_0 \vdash_{a_0} t : A_0$, then $\mathcal{X}, \mathcal{X}'|\Delta, \Delta'|\Gamma \vdash_a e[\frac{t}{\text{let } z \text{ be } u[\theta] \text{ in } z}] : A$.

PROOF.

We apply Lemma 1 multiple times to $\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A_0|\Gamma \vdash_a e : A$, to get $\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A_0, \Delta'|\Gamma \vdash_a e : A$. We then apply Lemma 2 to get $\mathcal{X}, \mathcal{X}'|\Delta, u : [\Gamma_0]_{a_0} A_0, \Delta'|\Gamma \vdash_a e : A$. Now we just Cut! $\mathcal{X}, \mathcal{X}'|\Delta, \Delta'|\Gamma_0 \vdash_{a_0} t : A_0$ into this to get $\mathcal{X}, \mathcal{X}'|\Delta, \Delta'|\Gamma \vdash_a e[\frac{t}{\text{let } z \text{ be } u[\theta] \text{ in } z}] : A$. □

Definition:

A goal refinement is the process of taking a proof $\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A_0|\Gamma \vdash_a e : A$ and constructing $\mathcal{X}, \mathcal{X}'|\Delta, \Delta'|\Gamma \vdash_a e[\frac{t}{u}] : A$, by building a term t such that $\mathcal{X}, \mathcal{X}'|\Delta, \Delta'|\Gamma_0 \vdash_{a_0} t : A_0$ (obtained by applying one of 14 refinement rules) and then substituting this term t for u in e. The rules available are dependent on the current state of the sequent and are presented below (the notation $a_0 = \{x_1, .., x_n\} \notin \mathcal{X}$ means that α_0 is not already set to something by $\mathcal{X}$):

1. $id(x)$ - available when $x : A_0 \in \Gamma_0$, $a_0 \subset_{\mathcal{X}} \{x, ...\}$, and $a_0 = \{x_1, .., x_n\} \notin \mathcal{X}$.

2. 1_R - available when $A_0 = 1$ and $a_0 = \{x_1, .., x_n\} \notin \mathcal{X}$.

3. $1_L(z)$ - available when $z : 1 \in \Gamma_0$, $a_0 \subset_{\mathcal{X}} \{z, ...\}$, and $a_0 = \{x_1, ..., x_n\} \notin \mathcal{X}$.

4. $\otimes_R$ - available when $A_0 = B \otimes C$.

5. $\otimes_L(x)$ - available when $x : B \otimes C \in \Gamma_0$, $a_0 \subset_{\mathcal{X}} \{x, ...\}$ and $a_0 = \{x_1, ..., x_n\} \notin \mathcal{X}$.

6. $\multimap_R$ - available when $A_0 = B \multimap C$.

7. $\multimap_L (f)$ - available when $f : B \multimap C \in \Gamma_0$, $a_0 \subset_{\mathcal{X}} \{f, ...\}$ and $a_0 = \{x_1, ..., x_n\} \notin \mathcal{X}$.

8. $\oplus_{R_1}$ - available when $A_0 = B \oplus C$.

9. $\oplus_{R_2}$ - available when $A_0 = B \oplus C$.

10. $\oplus_L(z)$ - available when $z : B \oplus C \in \Gamma_0$, $a_0 \subset_{\mathcal{X}} \{z, ...\}$ and $a_0 = \{x_1, ..., x_n\} \notin \mathcal{X}$.

11. $\&_R$ - available when $A_0 = B \,\&\, C$.

12. $\&_{L_1}(z)$ - available when $z : B \,\&\, C \in \Gamma_0$, $a_0 \subset_{\mathcal{X}} \{z, ...\}$ and $a_0 = \{x_1, ..., x_n\} \notin \mathcal{X}$.

13. $\&_{L_2}(z)$ - available when $z : B \,\&\, C \in \Gamma_0$, $a_0 \subset_{\mathcal{X}} \{z, ...\}$ and $a_0 = \{x_1, ..., x_n\} \notin \mathcal{X}$.

14. $MV(v)$ - available when $v \in \Delta$.

Theorem:

Every goal refinement in a term that typechecks under a consistent restriction context, results in a term that also typechecks under a consistent restriction context.

Proof.

We are given $\mathcal{X}|\Delta, u : [\Gamma_0]_{a_0} A_0|\Gamma \vdash_a e : A$ and we know that $\mathcal{X}$ is consistent. We want to refine the goal u, so we first create $\mathcal{X}, \mathcal{X}'|\Delta, \Delta'|\Gamma_0 \vdash_{a_0} t : A_0$. We proceed by a proof of reasoning by cases on the type of rules used to refine the goal. We

present the id, $\otimes_R$ and $\otimes_L$ rules.

id

The proof assistant only allows us to apply an id-rule on a variable x if a_0 has not already been set to something else inside of $\mathcal{X}$, and if x is contained in the upper bound multiset of variables that a_0 can use. In that case we get:

$$\frac{x : A_0 \in \Gamma_0 \qquad (a_0 = \{x\}) \in \mathcal{X}, (a_0 = \{x\})}{\mathcal{X}, (a_0 = \{x\}) | \Delta | \Gamma_0 \vdash_{a_0} x : A_0}\ \text{id}$$

Since $a_0 = \{x_1, ..., x_n\} \notin \mathcal{X}$, and x is a variable that can be used under a_0, adding the restriction $(a_0 = \{x\})$, keeps the context consistent.

$\otimes_{\boldsymbol{R}}$

We are only allowed to refine using $\otimes_R$ when $A_0 = B \otimes C$. Let $\mathcal{X}' = \{(a_0 = a_1 \cup a_2), (a_1' = \{z\}), (a_1 = a_1'[\frac{a_1[id]}{z}]), (a_2' = \{w\}), (a_2 = a_2'[\frac{a_2[id]}{w}])\}$ and $\Delta' = u_1 : [\Gamma_0]_{a_1} B, u_2 : [\Gamma_0]_{a_2} C$.

Then let $M =$

$$\frac{\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0 \vdash_{id} id : \Gamma_0 \qquad \dfrac{z : B \in \Gamma_0, z : B \qquad a_1' = \{z\} \in \mathcal{X}, \mathcal{X}'}{\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0, z : B \vdash_{a_1'} z : B} \qquad a_1 = a_1'[\frac{a_1[id]}{z}] \in \mathcal{X}, \mathcal{X}'}{\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0 \vdash_{a_1} \text{let } z \text{ be } u_1[id] \text{ in } z : B}\ MV$$

and $N =$

$$\frac{\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0 \vdash_{id} id : \Gamma_0 \qquad \dfrac{w : C \in \Gamma_0, w : C \qquad a_2' = \{w\} \in \mathcal{X}, \mathcal{X}'}{\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0, w : C \vdash_{a_2'} w : C} \qquad a_2 = a_2'[\frac{a_2[id]}{w}] \in \mathcal{X}, \mathcal{X}'}{\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0 \vdash_{a_2} \text{let } w \text{ be } u_2[id] \text{ in } w : C}\ MV$$

Then we have

$$\frac{M \qquad N \qquad a = a_1 \cup a_2 \in \mathcal{X}, \mathcal{X}'}{\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0 \vdash_{a_0} (\text{let } z \text{ be } u_1[id] \text{ in } z, \text{let } w \text{ be } u_2[id] \text{ in } w) : B \otimes C} \otimes_R$$

The context is still consistent because no matter what α is, one assignment that is true on all constraints is setting α_1 to be empty, while α_2 is all of α.

$\otimes_L$

The proof assistant only allows us to apply a left rule on a variable x if a_0 has not already been set to something else inside of $\mathcal{X}$, and if x is contained in the upper bound multiset of variables that a_0 can use and $x : B \otimes C \in \Gamma_0$. In that case we get:

Let $\mathcal{X}' = \{(a_0[\frac{\{x_1,x_2\}}{\{x\}}] = a_0'), (a_0' = a_0''[\frac{a_0'[id]}{z}]), (a_0'' = \{z\})\}$ and $\Delta' = u_1 : [\Gamma_0, x : B \otimes C, x_1 : B, x_2 : C]_{a_1} A_0$.

Then let $M =$

$$\frac{\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0, x_1, x_2 \vdash_{id} id : \Gamma_0, x_1, x_2 \qquad \dfrac{z : A_0 \in \Gamma_0, z : A_0, x_1 : B, x_2 : C \qquad a_0'' = \{z\} \in \mathcal{X}, \mathcal{X}'}{\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0, x_1, x_2, z : A_0 \vdash_{a_0''} z : A_0} \qquad a_0' = a_0''[\frac{a_0'[id]}{z}] \in \mathcal{X}, \mathcal{X}'}{\mathcal{X}, \mathcal{X}' | \Delta | \Gamma_0, x_1 : B, x_2 : C \vdash_{a_0'} \text{let } z \text{ be } u_1[id] \text{ in } z : A_0} MV$$

Then we have

$$\frac{M \qquad a_0[\frac{\{x_1,x_2\}}{\{x\}}] = a_0' \in \mathcal{X}, \mathcal{X}'}{\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0 \vdash_{a_0} \text{let } (x_1, x_2) \text{ be } x \text{ in } (\text{let } z \text{ be } u_1[id] \text{ in } z) : A_0} \otimes_L$$

Since $a_0 = \{x_1, ..., x_n\} \notin \mathcal{X}$, and x is a variable that can be used under a_0, adding the restriction that uses up x in a_0 by flipping it for x_1, x_2 in a_0', keeps the context consistent.

After we have completed $\mathcal{X}, \mathcal{X}' | \Delta, \Delta' | \Gamma_0 \vdash_{a_0} t : A_0$ we use Corollary on it and $\mathcal{X} | \Delta, u : [\Gamma_0]_{a_0} A_0 | \Gamma \vdash_a e : A$ and we are done. □

4.3. Implementation details.

In order to implement the proof assistant, we used OCaml. It became the language of choice because of its easy-to-define ADTs and its bigger popularity in the industry compared to Standard ML.

The project is broken down in multiple components:

1. A TermVar module used for storing and comparing Term, Meta and Resource variables.

2. A Syntax backbone module where types, terms and functions on them are defined.

3. A Typechecker module that is used to check if we have a valid term and a consistent context of restrictions that can identify all αs.

4. The main module which keeps the tables of contexts and runs the proof assistant.

When the program starts, the context that the user enters is parsed and saved into a hashtable that maps term variables to their types. The intended type entered by the user is also parsed and saved as the type of the first metavariable, together with the context hashtable and a new resource variable. That resource variable is then a part of an equation that says that it has to use up all of the resources from the entered context, and the equation is saved in the restrictions context.

After this we are in a loop whose guard checks if our term doesn't contain any metavariables in which case we exit the loop and are done if the term typechecks (with a consistent restrictions context).

If our term does contain metavariables, after the user selects a metavariable (goal) to work on, the proof assistant comes up with all of the useful information the user might need to construct a term: available variables and metavariables, type of the goal and applicable rules.

The refinement process works by replacing the metavariable in the term with the newly constructed term and updating the metavariable and restrictions context with the new metavariables and restrictions.

5. Conclusion

We have presented the process of building a proof assistant for propositional linear logic. In doing so, we have built two sequent calculi and showed their consistency. Because of the use of modal contexts and meta variables as well as the need for efficient splitting of linear contexts of variables, we introduced an implementation framework which delays important decisions like resource allocation. This allows the user to have more freedom in building a proof and adjust the resources available while building the proof. We chose linear logic as an example logic where these issues come up, but this framework can be used for many other substructural logics too.

This project can be further improved by adding types and terms for the quantifiable part of linear logic that we omitted, as well as dependent types. An in-editor implementation would also be a big step forward from the current terminal-based proof assistant.

Bibliography

[1] Gerhard Gentzen. Untersuchungen über das logische schließen. i. *Mathematische Zeitschrift*, 39(1):176–210, 1935.

[2] Jean-Yves Girard. Linear logic: Its syntax and semantics. In *Proceedings of the Workshop on Advances in Linear Logic*, pages 1–42, New York, NY, USA, 1995. Cambridge University Press.

[3] Georges Gonthier. *The Four Colour Theorem: Engineering of a Formal Proof*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[4] Joshua Seth Hodas. *Logic Programming in Intuitionistic Linear Logic: Theory, Design, and Implementation*. PhD thesis, Philadelphia, PA, USA, 1995. UMI Order No. GAX94-27546.

[5] J.S. Hodas and D. Miller. Logic programming in a fragment of intuitionistic linear logic. *Inf. Comput.*, 110(2):327–365, May 1994.

[6] William A. Howard. The formulas-as-types notion of construction. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, pages 479–490. Academic Press, 1980. Reprint of 1969 article.

[7] Robin Milner. Logic for computable functions: Description of a machine implementation. Technical report, Stanford, CA, USA, 1972.

[8] Aleksandar Nanevski, Frank Pfenning, and Brigitte Pientka. Contextual modal type theory. *ACM Trans. Comput. Logic*, 9(3):23:1–23:49, June 2008.

[9] Ulf Norell. *Towards a practical programming language based on dependent type theory*. PhD thesis, Department of Computer Science and Engineering, Chalmers University of Technology, SE-412 96 Göteborg, Sweden, September 2007.

[10] Anders Schack-nielsen and Carsten Schürmann. Linear contextual modal type theory. Technical Report TR-2011-151, IT University of Copenhagen, December 2011.