Logical Structures of the Catch and Throw Mechanism

(キャッチアンドスロー機構の論理的構造)

中 野　浩

Logical Structures of
the Catch and Throw Mechanism

(キャッチアンドスロー機構の論理的構造)

Hiroshi Nakano
Department of Applied Mathematics and Informatics
Ryukoku University

A Dissertation
Submitted to the Graduate School of
the University of Tokyo
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Science

February, 1995
(Revised May, 1995)

Abstract

The catch and throw mechanism is a programming construct for non-local exit. In practical programming, this mechanism plays an important role when programmers handle exceptional situations. In this thesis we give typing systems which capture the mechanism in the proofs-as-programs notion. The typing systems can be regarded as a constructive logic with facilities for exception handling, which includes inference rules corresponding to the operations of *catch* and *throw*. We show that we can actually regard their proofs as programs which make use of the catch and throw mechanism by a natural interpretation. On one hand the catch and throw mechanism provides only a restricted access to the current continuation, on the other hand its logic is still constructive, in contrast to the works due to Griffin and Murthy on more powerful facilities such as *call/cc* (call-with-current-continuation) of Scheme. We also capture the non-determinism introduced by the catch and throw mechanism in a consistent way.

Acknowledgements

I would like to thank Susumu Hayashi for a great number of helpful suggestions and invaluable encouragement on this work. This work has been deeply influenced by him in a number of aspects. He pointed out the connection between this work and the literature concerning variants of LJ and the treatment of *goto* in Hoare's logic. I would like to thank Satoru Takasu, who enlightened me on the fundamental notions of constructive programming. I would like to thank Masami Hagiya for his valuable comments and helpful support on this thesis. I would also thank Takayasu Ito, Masahiko Sato, Makoto Tatsuta, Satoshi Kobayashi, and many other people who contributed directly or indirectly to the development of this thesis. Finally I wish to thank Hiromi Nakano, my wife, for her patient encouragement and support during this work.

Contents

1 Introduction **1**

1.1 Backgrounds . . . 1

1.2 The catch and throw mechanism . . . 2

1.3 Overview of the thesis . . . 4

1.4 Related works . . . 5

2 A typing system for the catch and throw mechanism **7**

2.1 A calculus with the catch and throw mechanism . . . 7

2.1.1 Syntax . . . 7

2.1.2 Operational semantics of the calculus . . . 8

2.2 The typing system . . . 11

2.3 A realizability interpretation . . . 14

2.3.1 The realizability . . . 15

2.3.2 Soundness . . . 16

3 The conventional implementation **20**

3.1 Definition of the machine . . . 20

3.2 Validity of the machine . . . 23

3.3 Realizability by the abstract machine . . . 39

4 The typing system as a logic **40**

4.1 A sequent calculus style formulation . . . 40

4.2 The logical meaning of the new connective . . . 42

4.3 The catch and throw mechanism as structural rules . . . 43

4.4 The restriction on the right implication rule . . . 43

4.5 The cut-elimination theorem . . . 44

5 A natural extension with a non-determinism **49**

5.1 A non-determinism by the catch and throw mechanism . . . 49

5.2 A calculus with a non-determinism . . . 49

5.2.1 Operational semantics . . . 50

5.3 Basic properties of the calculus . . . 51

5.4 The typing system $L_{c/t}$ . . . 52

5.4.1 Syntax of typing judgements . . . 52

5.4.2 $L_{c/t}$ . . . 53

5.4.3 Basic properties of $L_{c/t}$ . . . 55
5.5 The subject reduction property of $L_{c/t}$ . . . 57
5.6 $L_{c/t}$ as a logic . . . 60

6 A term model for the extended system **62**
6.1 Term models . . . 62
6.2 An admissible frame . . . 73
6.3 Strong normalizability and normal forms . . . 84
6.4 Realizability interpretation of $L_{c/t}$ . . . 85

7 Concluding remarks **86**

Chapter 1

Introduction

1.1 Backgrounds

It is well known that constructive proofs can be regarded as computer programs by a notion called "proofs as programs". By this notion, we can extract executable programs from constructive proofs of sentences that correspond to the specifications of the programs by a "realizability interpretation" [16]. The same schema can be observed in the area of constructive type theories, in which types specify what programs do. The notion of "proofs as programs" is also called "formula as types", "propositions as types" or the Curry-Howard isomorphism [14], and is summarized as the following correspondences.

Computer programming	Constructive logic or type theory
programs	proofs
specifications	formulas or types
program development	theorem proving
programmers	mathematicians or ...

This paradigm provides a theoretical basis for a formal method of computer programming, in which programmers construct formal proofs of theorems that specify what the programs do, and target programs in some form are automatically extracted from the formal proofs. The correctness of the programs relative to their specification, that is, the theorems, is defined by a certain interpretation of formulas of the formal system, and is assured by the soundness metatheorem for the formal system with respect to the interpretation. In other words, programmers simultaneously construct and verify the computer programs in this paradigm.

In the last decade, many works have been intensively done both in practical and in theoretical approaches [5, 6, 12, 13, 19, 26, 35, 36], in which the most of their attention has been concentrated on the area that can be regarded as an application of the standard constructive logic, because we had already have rich results on the constructive logic itself, and the conventional constructive logics really have enough strength with respect to the class of provable theorems, that is, the class of realizable specifications. However, from the viewpoint of practical programming, some aspects of the logical activities of programmers have not been captured in this paradigm. For example, the class of proofs, that is, the class of available programs is also important for the programmers as well as the class of provable theorems, but is not ever discussed intensively. Actually, many practical programming languages provide some additional programming facilities to programmers in order to extend the class of programs. Although these facilities do not extend the class of realizable specifications, that is, we can construct equivalent programs without such additional facilities, they have important roles for practical program development. The main aim of this thesis is to capture one of such programming facilities in the notion of "proofs as programs".

1.2 The catch and throw mechanism

The catch and throw mechanism is a programming facility for non-local exit. We can find examples of the mechanism in some practical programming languages such as C [15] and Common Lisp [32]. We give some primary introduction about this mechanism by taking Common Lisp as an example. In the case of Common Lisp, `(catch` *tag form*`)` is a special form that serves as a target of transfer of control by another special form `(throw` *tag form*`)`. At the evaluation of `(catch` *tag form*`)`, a catcher marked with the *tag* is established, and the *form* is then evaluated. The result of the *form* is returned from the `catch`, except that if a `throw` special form with the same *tag* is executed during the evaluation, then the evaluation is immediately aborted and the `catch` returns a value specified by the *form* of the `throw`. For example, the evaluation process of `(+ (catch 'u (+ 2 (+ 3 5))) 8)` proceeds as follows.

```
Eval: (+ (catch 'u (+ 2 (+ 3 5))) 8)
  :  Eval: (catch 'u (+ 2 (+ 3 5)))
  :    :  Eval: 'u
  :    :  Return: u
  :    :  Eval: (+ 2 (+ 3 5))
  :    :    :  Eval: 2
  :    :    :  Return: 2
  :    :    :  Eval: (+ 3 5)
  :    :    :    :  Eval: 3
  :    :    :    :  Return: 3
  :    :    :    :  Eval: 5
  :    :    :    :  Return: 5
  :    :    :    :  Apply: + to 3 and 5
  :    :    :  Return: 8
  :    :    :  Apply: + to 2 and 8
  :    :  Return: 10
  :  Return: 10
  :  Eval: 8
  :  Return: 8
  :  Apply: + to 10 and 8
Return: 18
```

On the other hand, the evaluation process of `(+ (catch 'u (+ 2 (+ 3 (throw 'u 5)))) 8)` proceeds as follows.

```
Eval: (+ (catch 'u (+ 2 (+ 3 (throw 'u 5)))) 8)
    Eval: (catch 'u (+ 2 (+ 3 (throw 'u 5))))
        Eval: 'u
        Return: u
        Eval: (+ 2 (+ 3 (throw 'u 5)))
            Eval: 2
            Return: 2
            Eval: (+ 3 (throw 'u 5))
                Eval: 3
                Return: 3
                Eval: (throw 'u 5)
                    Eval: 'u
                    Return: u
                    Eval: 5
                    Return: 5
                    Apply: throw to u and 5
    Return: 5
    Eval: 8
    Return: 8
    Apply: + to 2 and 8
Return: 10
```

In practical programming, the catch and throw mechanism plays an important role when programmers handle exceptional situations. Suppose, for example, we have to construct a program P with a specification represented by a sequent $\Gamma \to C \vee E$ of LJ combining three subprograms $Q : \Gamma \to A \vee E$, $R : \Gamma A \to B \vee E$ and $S : \Gamma B \to C \vee E$, where C is the normal output for the input Γ, and E is an error signal which denotes that there is something wrong in the input. The specifications of Q, R and S say that such errors may be detected in the execution of these subprograms. In this situation, the construction of P in LJ would be as follows,

$$\dfrac{\begin{array}{c}\vdots\, Q\\ \Gamma \to A \vee E\end{array} \qquad \dfrac{\dfrac{\begin{array}{c}\vdots\, R\\ \Gamma A \to B \vee E\end{array} \qquad \dfrac{\begin{array}{c}\vdots\, \mathcal{S}\\ \Gamma B \to C \vee E\end{array} \quad \dfrac{\dfrac{}{E \to E}\,(init)}{E \to C \vee E}\,(\to\vee)}{\Gamma B \vee E \to C \vee E}\,(\vee\to)}{\Gamma A \to C \vee E}\,(cut) \qquad \dfrac{\dfrac{}{E \to E}\,(init)}{E \to C \vee E}\,(\to\vee)}{\Gamma A \vee E \to C \vee E}\,(\vee\to)}{\Gamma \to C \vee E}\,(cut),$$

where applications of structural rules are omitted. The constructed program P would work as follows. The program P first calls the subprogram Q and gets its return value, then checks whether it denotes an error or not. If not, P calls the subprogram R with that value and gets its return value. P again checks the value and calls S if it does not denote an error. Eventually, P returns the value returned by S. If P detects an error in the values returned by Q or S, it immediately returns a value denoting an error. We can find an inefficiency that whenever P gets values from the subprograms it must check whether they denote an error or not. This is often found in practical programming without the catch and throw mechanism. If the mechanism is available, programmers can concentrate on the main stream of programming as if no exceptional situation can arise, since error signals are passed through ordinary language constructs. Following the proofs-as-programs notion in the opposite direction, we can find that the problem comes from the restriction of LJ that only one formula is admissible in the right-hand side of sequents. If the restriction is dropped as in LK, we can construct P' of a specification $\Gamma \rightarrow C\,E$ from subprograms $Q' : \Gamma \rightarrow A\,E$, $R' : \Gamma\,A \rightarrow B\,E$ and $S' : \Gamma\,B \rightarrow C\,E$ as follows,

$$\dfrac{\begin{array}{c}\vdots\, Q' \\ \Gamma \rightarrow A\,E\end{array} \qquad \dfrac{\begin{array}{c}\vdots\, R' \\ \Gamma\,A \rightarrow B\,E\end{array} \qquad \begin{array}{c}\vdots\, S' \\ \Gamma\,B \rightarrow C\,E\end{array}}{\Gamma\,A \rightarrow C\,E}\,(cut)}{\Gamma \rightarrow C\,E}\,(cut),$$

where structural rules are again omitted. The proof is much simpler than the previous one and easy to develop. The point is that exceptional conclusions are admitted besides the main conclusion and we can proceed the proof construction as if they do not exist. It reflects the programmer's reasoning behind the catch and throw mechanism well. Of course, we must justify such a logic constructively so that correct programs can be extracted from the proofs.

In this thesis, we present such an attempt to extract a logical structure from the programmer's reasoning concerning exception handling by the catch and throw mechanism, and capture the mechanism in the notion of "proofs as programs" by constructive frame works and their realizability interpretations.

1.3 Overview of the thesis

Chapter 2 gives a typing system for a simple programming language equipped with the catch and throw mechanism. The aim of this chapter is to capture the mechanism with a fixed evaluation strategy, the call-by-value strategy. Section 2.1 introduces a calculus with the catch and throw mechanism, and gives its operational semantics by a set of reduction rules in the manner of Felleisen *et al.* [7]. This semantics should be a natural translation of the standard operational semantics of the mechanism in practical programming languages. Section 2.2 gives a typing system $L_{c/t}^{\mathrm{CBV}}$ of the calculus, and Section 2.3 gives a realizability interpretation of the typing system and shows how the catch and mechanism can be captured by proving the soundness theorem.

Chapter 3 introduces an abstract stack machine to imitate the standard implementation of the catch and throw mechanism in practical programming languages, and shows that the semantics defined by this machine is equivalent to the one given in Section 2.1. Another realizability interpretation of $L_{c/t}^{\mathrm{CBV}}$ defined directly by the abstract machine is also given.

Chapter 4 discusses the formal system regarding it as a logic. We show that $L_{c/t}^{\mathrm{CBV}}$ is a conservative extension of the propositional fragment of the standard intuitionistic logic such as Gentzen's LJ or NJ. We reformulate $L_{c/t}^{\mathrm{CBV}}$ into a sequent calculus style formal system in order to clarify the difference between $L_{c/t}^{\mathrm{CBV}}$ and the standard formulations of classical and intuitionistic logic. The cut-elimination theorem of the sequent calculus style reformulation is also given.

Chapter 5 deals with the non-determinism introduced by the catch and throw mechanism. The reduction rules of the calculus given in Chapter 2 is naturally extended to capture this non-determinism, in which any evaluation strategy should be allowed. The typing system of the calculus is also extended. We show that although the well-typed terms of the system do not have the Church-Rosser property, they have the subject reduction property instead.

Chapter 6 is devoted to construction of a term model of the extended system introduced in the previous chapter. Unfortunately, the standard method is not enough to this construction, and a set of terms that has a certain property is required for the domain of interpretation. By this term model, we show the strong normalizability of the well-typed terms and the soundness of the extended typing system relative to an extended realizability interpretation.

1.4 Related works

From a computational point of view, the catch and throw mechanism is just a subcase of more powerful facilities for control such as *call/cc* of Scheme and Felleisen's $\mathcal{C}$ operator [7], and the relation between such facilities and the computational meaning of classical logic has been investigated in various ways [3, 11, 21, 20, 22, 27, 28, 29, 31], where the computational behavior of the facilities is captured as translation processes of classical proofs into intuitionistic ones [8].

Murthy showed that classical formal proofs can be regarded as programs with such control facilities [21, 20], where Felleisen's $\mathcal{C}$ operator corresponds to the following inference rule of classical logic.

$$\frac{\neg\neg A}{A}\ (\mathcal{C})$$

By a simple modification of Freidman's work [8], he showed that classical proofs of a formula A can be translated into intuitionistic proofs of $(A^{\Phi} \supset \Phi) \supset \Phi$ for any formula Φ, where

$$\begin{array}{rcll}
A^{\Phi} & = & A & (A \text{ is atomic}) \\
(\neg A)^{\Phi} & = & A^{\Phi} \supset \Phi & \\
(A \wedge B)^{\Phi} & = & A^{\Phi} \wedge B^{\Phi} & \\
(A \vee B)^{\Phi} & = & A^{\Phi} \vee B^{\Phi} & \\
(A \supset B)^{\Phi} & = & A^{\Phi} \supset (B^{\Phi} \supset \Phi) \supset \Phi & \\
(\forall x.A)^{\Phi} & = & \forall x.((A^{\Phi} \supset \Phi) \supset \Phi) & \\
(\exists x.A)^{\Phi} & = & \exists x.A^{\Phi}. &
\end{array}$$

Note that if Φ does not include any $\neg$, $\supset$ or $\forall$, then we can translate classical proofs of Φ into intuitionistic ones since $\Phi^{\Phi} = (\Phi \supset \Phi) \supset \Phi$ in this case. This translation of formulas that maps A to A^{Φ} corresponds to the continuation-passing-style translation [30] of programs of type A, and the computational meaning of $\mathcal{C}$ operator can be regarded as a mechanism that transforms intuitionistic proofs of $((\neg\neg A)^{\Phi} \supset \Phi) \supset \Phi$ into ones of $(A^{\Phi} \supset \Phi) \supset \Phi$.

However, from the viewpoint of formal method for computer programming, only a restricted class of formulas, for example, Π^0_2 sentences, are allowed for the specification of programs in order to assure the total correctness of the programs in these classical frameworks. On the other hand, any of such restrictions is not required for the frameworks presented in this thesis, that is, any formula is allowed as a specification of programs. More importantly, the main aim of our work is to capture the logic behind the use of such facilities in practical programming rather than to capture their computational behavior. We summarize the basic properties of our frameworks in comparison among others in the following table.

framework	logic	trick	proof as programs	non-determinism
Murthy [20]	classical	$\mathcal{C}$ operator	restricted	no
$\lambda\mu$-calculus [29]	classical	μ operator	restricted	no
LC [9]	classical	involutive negation	restricted	no
λ^{Sym}_{Prop} [3]	classical	involutive negation	restricted	yes
$L^{\mathrm{CBV}}_{c/t}$	intuitionistic	catch/throw	yes	no
$L_{c/t}$	intuitionistic	catch/throw	yes	yes

Chapter 2

A typing system for the catch and throw mechanism

2.1 A calculus with the catch and throw mechanism

In this section we introduce a simple programming language equipped with the catch and throw mechanism, and give its operational semantics.

2.1.1 Syntax

We first give the syntax of the language.

Definition 2.1.1 (Constants and variables) We assume that the following disjoint sets of individual constants, individual variables and tag variables are given. The syntax of the calculus is defined relatively to these three sets.

$$\begin{array}{lcl} \mathit{Const} & : & \text{a set of individual constants } c, d, \ldots. \\ \mathit{Var} & : & \text{a countably infinite set of individual variables } x, y, z, \ldots. \\ \mathit{Tvar} & : & \text{a countably infinite set of tag variables } u, v, w, \ldots. \end{array}$$

Definition 2.1.2 (Terms) We define a set *Term* of terms by

$$\begin{array}{rcl} \mathit{Term} & ::= & \mathit{Const} \quad | \quad \mathit{Var} \\ & | & \mathbf{let}\ \mathit{Var}{=}\mathit{Term}.\ \mathit{Term} \\ & | & \mathbf{throw}\ \mathit{Tvar}\ \mathit{Term} \quad | \quad \mathbf{catch}\ \mathit{Tvar}\ \mathit{Term} \\ & | & \lambda\, \mathit{Var}.\, \mathit{Term}^* \quad | \quad \mathit{Term}\, \mathit{Term} \\ & | & <\mathit{Term},\ \mathit{Term}> \quad | \quad \mathbf{proj}_1\, \mathit{Term} \quad | \quad \mathbf{proj}_2\, \mathit{Term} \\ & | & \mathbf{inj}_1\, \mathit{Term} \quad | \quad \mathbf{inj}_2\, \mathit{Term} \quad | \quad \mathbf{case}\ \mathit{Term}\ \ \mathit{Var}.\mathit{Term}\ \ \mathit{Var}.\mathit{Term} \\ & | & \kappa\, \mathit{Tvar}.\ \mathit{Term} \quad | \quad \mathit{Term}\, \mathit{Tvar}\ , \end{array}$$

where Term^* stands for the set of terms that have no free occurrence of tag variables. We use $M, N, K, L, \ldots$ to denote terms.

Free and bound occurrences of variables are defined in the standard manner. We regard a tag variable u as bound in **catch** $u\ M$ and $\kappa\, u.\, M$. We denote the set of individual and tag variables occurring freely in M by $FIV(M)$ and $FTV(M)$, respectively.

Example 2.1.3 (Terms)

$$\lambda\, x.\, \mathbf{case}\ x\ y.(\mathbf{inj_2}\ y)\ z.(\mathbf{inj_1}\ z)$$

$$\mathbf{catch}\ u\ ((\kappa\, v.\, \mathbf{proj_1} <x,\ \mathbf{throw}\ v\ y>)\ u)$$

Note that we have a restriction on terms of the form $\lambda\, x.\, M$. For example, $\lambda\, x.\, \mathbf{throw}\ u\ x$ is not a term since it has a free occurrence of a tag variable u.

The terms of the forms **throw** *Tvar Term* and **catch** *Tvar Term* provide the catch and throw mechanism of the language. Roughly, these terms correspond to `throw` and `catch` of Common Lisp [32], respectively. The terms $\kappa\ Tvar.Term$ and $Term\,Tag$ are used for tag-abstraction and tag-instantiation, respectively. These terms provide a way of passing tags. Note that tags can not be passed by usual lambda-abstractions and applications since they are not terms.

We also define the alpha-convertibility in the standard manner where we allow renaming of bound tag variables as well as bound individual variables. Hereafter, we identify terms by this alpha-convertibility. We use $M[N_1/x_1, \ldots, N_n/x_n]$ to denote the term obtained from a term M by substituting $N_1, \ldots, N_n$ for each free occurrence of individual variables $x_1, \ldots, x_n$, respectively, with alpha-conversion for avoiding capture of free variables. Similarly, we also use $M[v_1/u_1, \ldots, v_n/u_n]$ to denote the term obtained from M by substituting $v_1, \ldots, v_n$ for each free occurrence of tag variables $u_1, \ldots, u_n$, respectively. Note that in the case that $N_1, \ldots, N_n$ have free tag variables, $M[N_1/x_1, \ldots, N_n/x_n]$ may not be a well-formed term, because $N_1, \ldots, N_n$ can introduce some free tag variables into lambda-abstractions.

Proposition 2.1.4 *Let M be a term, and let $x_1, \ldots, x_m$ and $u_1, \ldots, u_n$ be sequences of distinct individual and tag variables, respectively. If $N_1, \ldots, N_m$ are closed terms and $v_1, \ldots, v_n$ are tag variables, then $M[N_1/x_1, \ldots, N_m/x_m, v_1/u_1, \ldots, v_n/u_n]$ is a term.*

Proof. Obvious from the definition of *Term*. □

2.1.2 Operational semantics of the calculus

We define a call-by-value evaluator of terms to give an intuitive semantics of this calculus. The evaluator is defined in terms of evaluation contexts and a set of rewriting rules for terms. The basic idea is due to Felleisen *et al.* [7]. In Chapter 3 we will give another operational semantics by an abstract machine which imitates the standard implementation of the catch/throw mechanism in some practical programming languages.

Definition 2.1.5 (Values) We define a set *Val* of closed terms as follows.

$$\begin{array}{rcl} Val & ::= & Const \\ & | & \lambda\, Var.Term \quad | \quad <Val,\ Val> \\ & | & \mathbf{inj_1}\, Val \quad | \quad \mathbf{inj_2}\, Val \\ & | & \kappa\, Tvar.\ Val \quad | \quad \kappa\, Tvar.\ \mathbf{throw}\ Tvar\ Val\ . \end{array}$$

Elements of *Val* are called *values*, and we use $V, V', W, W', \ldots$ to denote values.

Note that any value must be a closed term. Therefore any value of the form

$$\kappa\, \mathit{Tvar}.\, \mathbf{throw}\ \mathit{Tvar}\ \mathit{Val}$$

must be $\kappa\, u.\, \mathbf{throw}\ u\ V$ for some tag variable u and some value V.

Example 2.1.6 (Values)

$$\lambda\, x.\, (\lambda\, y.\, y)\, x$$

$$<\mathbf{inj_1}\, c_1,\ \kappa\, u.\, \mathbf{throw}\ u\ c_2>$$

Definition 2.1.7 (Evaluation contexts) We define a set C of pseudo terms which have a hole, denoted by $*$, in them as follows.

$$\begin{array}{lcl} C & ::= & * \\ & | & \mathbf{let}\ \mathit{Var}{=}C.\ \mathit{Term} \\ & | & \mathbf{throw}\ \mathit{Tvar}\ C \quad | \quad \mathbf{catch}\ \mathit{Tvar}\ C \\ & | & C\, \mathit{Term} \quad | \quad \mathit{Val}\, C \\ & | & <C,\ \mathit{Term}> \quad | \quad <\mathit{Val},\ C> \quad | \quad \mathbf{proj_1}\, C \quad | \quad \mathbf{proj_2}\, C \\ & | & \mathbf{inj_1}\, C \quad | \quad \mathbf{inj_2}\, C \quad | \quad \mathbf{case}\ C\ \mathit{Var}.\mathit{Term}\ \mathit{Var}.\mathit{Term} \\ & | & \kappa\, \mathit{Tvar}.\, C \quad | \quad C\, \mathit{Tag}. \end{array}$$

Elements of C are called *evaluation contexts.* We use $\mathcal{C}$, $\mathcal{C}'$, ... to denote evaluation contexts. We use $\mathcal{C}[M]$ to denote the term obtained from $\mathcal{C}$ by replacing the hole $*$ by a term M.

By the definition, no evaluation context captures free individual variables placed at the hole, but it may capture free tag variables at the hole. For example, if $\mathcal{C} = \mathbf{catch}\ u\ *$ and $M = \mathbf{throw}\ u\ N$, then $\mathcal{C}[M] = \mathbf{catch}\ u\ (\mathbf{throw}\ u\ N)$.

Example 2.1.8 (Evaluation contexts)

$$(\lambda\, y.\, M\, (\mathbf{proj_1}\, y))\, <*,\ N>$$

$$\mathbf{catch}\ u\ (\mathbf{inj_1}\, (\mathbf{throw}\ u\ *))$$

Proposition 2.1.9 *If $\mathcal{C}$ and $\mathcal{C}'$ are evaluation contexts, then $\mathcal{C}[\mathcal{C}']$ is also an evaluation context.*

Proof. Obvious from the definition of evaluation contexts. □

Proposition 2.1.10 *If a term M can be written as $M = \mathcal{C}[\mathbf{throw}\ u\ V]$, where V is a value and $\mathcal{C}$ does not capture u, then the combination of such $\mathcal{C}$ and* **throw** *$u\ V$ is unique.*

Proof. By induction on the structure of $\mathcal{C}$. Note that no value has free occurrences of tag variables.
□

Definition 2.1.11 (Rewriting rules) The call-by-value rewriting $\underset{\text{CBV}}{\longmapsto}$ is defined by the following rules, where $\mathcal{C}$ is an arbitrary evaluation context such that $\mathcal{C} \neq *$ and $\mathcal{C}$ does not capture u. Note that none of redexes is a value.

$$
\begin{array}{rcl}
\mathbf{catch}\ u\ V & \underset{\text{CBV}}{\longmapsto} & V \\
\mathbf{catch}\ u\ \mathcal{C}[\mathbf{throw}\ u\ V] & \underset{\text{CBV}}{\longmapsto} & \mathbf{catch}\ u\ (\mathbf{throw}\ u\ V) \\
\mathbf{catch}\ u\ (\mathbf{throw}\ u\ V) & \underset{\text{CBV}}{\longmapsto} & V \\
\mathbf{let}\ x{=}V.\ M & \underset{\text{CBV}}{\longmapsto} & M[V/x] \\
(\lambda\, x.\, M)\, V & \underset{\text{CBV}}{\longmapsto} & M[V/x] \\
\kappa\, u.\, \mathcal{C}[\mathbf{throw}\ u\ V] & \underset{\text{CBV}}{\longmapsto} & \kappa\, u.\, \mathbf{throw}\ u\ V \\
(\kappa\, u.\, V)\, v & \underset{\text{CBV}}{\longmapsto} & V \\
(\kappa\, u.\, \mathbf{throw}\ u\ V)\, v & \underset{\text{CBV}}{\longmapsto} & \mathbf{throw}\ v\ V \\
\mathbf{proj_1} <V,\ W> & \underset{\text{CBV}}{\longmapsto} & V \\
\mathbf{proj_2} <V,\ W> & \underset{\text{CBV}}{\longmapsto} & W \\
\mathbf{case}\ (\mathbf{inj_1}\ V)\ x.M\ y.N & \underset{\text{CBV}}{\longmapsto} & M[V/x] \\
\mathbf{case}\ (\mathbf{inj_2}\ V)\ x.M\ y.N & \underset{\text{CBV}}{\longmapsto} & N[V/y]
\end{array}
$$

Proposition 2.1.12 *Let M be a term. If $M \underset{\text{CBV}}{\longmapsto} N$ for some N, then N is unique.*

Proof. Obvious from the definition of $\underset{\text{CBV}}{\longmapsto}$. □

Definition 2.1.13 (Evaluation steps) A call-by-value evaluation step $\underset{\text{CBV}}{\longrightarrow}$ is defined by

$$\mathcal{C}[M] \underset{\text{CBV}}{\longrightarrow} \mathcal{C}[N] \qquad \text{if and only if} \qquad M \underset{\text{CBV}}{\longmapsto} N.$$

Example 2.1.14 Let M be a term, and let V and V' be values.

$$
\begin{array}{rcl}
(\mathbf{catch}\ u\ ((\lambda\, z.\, M)\,(\mathbf{throw}\ u\ (\lambda\, x.\, x))))\ V & \underset{\text{CBV}}{\longrightarrow} & (\mathbf{catch}\ u\ (\mathbf{throw}\ u\ (\lambda\, x.\, x)))\ V \\
& \underset{\text{CBV}}{\longrightarrow} & (\lambda\, x.\, x)\ V \\
& \underset{\text{CBV}}{\longrightarrow} & V \\
\mathbf{catch}\ u\ (\mathbf{proj_2} <V,\ V'>) & \underset{\text{CBV}}{\longrightarrow} & \mathbf{catch}\ u\ V' \\
& \underset{\text{CBV}}{\longrightarrow} & V' \\
\mathbf{catch}\ u\ (((\lambda\, x.\, \kappa\, v.\, \mathbf{throw}\ v\ x)\ V)\ u) & \underset{\text{CBV}}{\longrightarrow} & \mathbf{catch}\ u\ ((\kappa\, v.\, \mathbf{throw}\ v\ V)\ u) \\
& \underset{\text{CBV}}{\longrightarrow} & \mathbf{catch}\ u\ (\mathbf{throw}\ u\ V) \\
& \underset{\text{CBV}}{\longrightarrow} & V
\end{array}
$$

Let $\overset{*}{\underset{\text{CBV}}{\longrightarrow}}$ be the transitive and reflexive closure of the relation $\underset{\text{CBV}}{\longrightarrow}$. We write $M \underset{\text{CBV}}{\succ} N$ if N is a normal form of M w.r.t. $\underset{\text{CBV}}{\longrightarrow}$, that is, $M \overset{*}{\underset{\text{CBV}}{\longrightarrow}} N$ and $N \underset{\text{CBV}}{\not\longmapsto} K$ for any K.

Proposition 2.1.15 *If $M \underset{\text{CBV}}{\longrightarrow} N$, then $\mathcal{C}[M] \underset{\text{CBV}}{\longmapsto} \mathcal{C}[N]$ for any evaluation context $\mathcal{C}$.*

Proof. Obvious from the definitions of $\underset{\text{CBV}}{\longrightarrow}$ and evaluation contexts. □

Proposition 2.1.16 *Let $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ be sequences of individual variables, and suppose that $x_1, \ldots, x_n$ are distinct. If $M \xrightarrow[\text{CBV}]{} N$, then $M[y_1/x_1, \ldots, y_n/x_n] \xrightarrow[\text{CBV}]{} N[y_1/x_1, \ldots, y_n/x_n]$. Similarly, let $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$ be sequences of tag variables, and suppose that $u_1, \ldots, u_n$ are distinct. If $M \xrightarrow[\text{CBV}]{} N$, then $M[v_1/u_1, \ldots, v_n/u_n] \xrightarrow[\text{CBV}]{} N[v_1/u_1, \ldots, v_n/u_n]$.*

Proof. Obvious from the definition of $\xrightarrow[\text{CBV}]{}$. □

Proposition 2.1.17 *If M is not a normal form, then M can be written in a unique way as $M = \mathcal{C}[N]$ for some evaluation context $\mathcal{C}$ and some redex N, i.e., $N \underset{\text{CBV}}{\longmapsto} K$ for some K.*

Proof. Since M is not normal, there exists at least one combination of such $\mathcal{C}$ and N. We can show the uniqueness by induction on the structure of $\mathcal{C}$. □

Therefore, if $M \underset{\text{CBV}}{\succ} N$, then N is unique. But note that N may not be a value even if $M \underset{\text{CBV}}{\succ} N$. For example, **throw** $u\ V \underset{\text{CBV}}{\succ}$ **throw** $u\ V$, but **throw** $u\ V$ is not a value.

Proposition 2.1.18 *Every value is a normal form.*

Proof. Obvious from the definition of $\xrightarrow[\text{CBV}]{}$. □

Proposition 2.1.19 *Let V be a value. Let u and C be a tag variable and an evaluation context, respectively. If $\mathcal{C}$ does not capture u, then $\mathcal{C}[$**throw** $u\ V]$ is a normal form.*

Proof. By induction on the form of $\mathcal{C}$. Note that $\mathcal{C}[$**throw** $u\ V]$ is not a value, because it is not a closed term. □

2.2 The typing system

We introduce a typing system $L_{c/t}^{\text{CBV}}$ of the language given in the previous section, which can be regarded as a logical system that captures the catch/throw mechanism.

Definition 2.2.1 (Type expressions) We have five kinds of *type expressions* in $L_{c/t}^{\text{CBV}}$ as follows.

$$\begin{array}{rcl} A & : & \text{atomic type expression} \\ A \wedge B & : & \text{conjunction} \\ A \vee B & : & \text{disjunction} \\ A \supset B & : & \text{implication} \\ A \triangleleft B & : & \text{exception} \end{array}$$

Type expressions are also called *formulas*.

The last one is introduced to handle the catch/throw mechanism and represents another kind of disjunction. It corresponds to the type of tag-abstractions, i.e., terms of the form $\kappa\, Tvar.Term$. We give a precise meaning to the connective $\triangleleft$ by a realizability interpretation later.

Definition 2.2.2 (Type-contexts) An *individual type-context*, or an individual context for short, is a finite mapping which assigns a type expression, i.e., a formula, to each individual variable in its domain. We use $\{x_1 : A_1, \ldots, x_m : A_m\}$ to denote an individual type-context whose domain is $\{x_1, \ldots, x_m\}$ and which assigns A_i to x_i for any i, where $A_1, \ldots, A_m$ are type expressions, and $x_1, \ldots, x_m$ are distinct individual variables. A *tag type-context*, or a *tag context* for short, is similarly defined as a finite mapping which assigns a type expression to each tag variable in its domain. We use $\{u_1 : B_1, \ldots, u_n : B_n\}$ to denote a tag type-context, where $u_1, \ldots, u_n$ are distinct tag variables.

Definition 2.2.3 (Typing judgement) Let Γ be an individual type-context, and Δ a tag type-context. Let M be a term, and C a type expression. *Typing judgements* have the following form.

$$\Gamma \vdash M : C\,;\ \Delta.$$

A typing judgement $\{x_1 : A_1, \ldots, x_m : A_m\} \vdash M : C\,; \{u_1 : E_1, \ldots, u_n : E_n\}$ roughly says that when we execute the program M supplying values of the types $A_1 \ldots A_m$ for the corresponding free variables $x_1, \ldots, x_m$ of the program, it normally returns a value of the type C, otherwise the program exits with a value which belongs to one of the types $E_1 \ldots E_n$. Let us explain by example. If we have a derivation of a typing judgement $\{x : A, y : B\} \vdash M : C\,;\ \{\}$, then

1. $FIV(M) \subset \{x, y\}$, and
2. if K and L satisfy the specifications A and B, respectively, then the evaluation of $M[K/x, L/y]$ terminates with a value that satisfies the specification C.

Note that this corresponds to the standard interpretation of simply typed lambda calculus. On the other hand, $\{x : A, y : B\} \vdash M : C\,;\ \{u : E\}$ says that

1. $FIV(M) \subset \{x, y\}$,
2. $FTV(M) \subset \{u\}$, and
3. if K and L satisfy the specifications A and B, respectively, then
    a. the evaluation of $M[K/x, L/y]$ terminates with a value that satisfies the specification C, or
    b. the evaluation of $M[K/x, L/y]$ causes a throw-operation of a value to the tag u, which is not caught by catch-terms, and the thrown value satisfies the specification E.

That is, there are two possible results of the evaluation of M in the second example. The precise meaning of typing judgements is given in the next section by a realizability interpretation.

Definition 2.2.4 (Typing rules) The inference rules of $L^{\text{CBV}}_{c/t}$ are as follows.

$$\frac{}{\Gamma \cup \{x : A\} \vdash x : A\,;\ \Delta}\ (var)$$

$$\frac{\Gamma \vdash N : A\,;\ \Delta \quad \Gamma \cup \{x : A\} \vdash M : C\,;\ \Delta}{\Gamma \vdash \mathbf{let}\ x{=}N.\ M : C\,;\ \Delta}\ (let)$$

$$\frac{\Gamma \vdash M : E\,;\ \Delta}{\Gamma \vdash \mathbf{throw}\ u\ M : A\,;\ \Delta \cup \{u : E\}}\ (throw) \qquad \frac{\Gamma \vdash M : A\,;\ \Delta \cup \{u : A\}}{\Gamma \vdash \mathbf{catch}\ u\ M : A\,;\ \Delta}\ (catch)$$

$$\frac{\Gamma \vdash M : A\,;\ \Delta \quad \Gamma \vdash N : B\,;\ \Delta}{\Gamma \vdash {<}M,\ N{>} : A \wedge B\,;\ \Delta}\ (\wedge\text{-I})$$

$$\frac{\Gamma \vdash M : A \wedge B\,;\ \Delta}{\Gamma \vdash \mathbf{proj_1}\, M : A\,;\ \Delta}\ (\wedge_1\text{-E}) \qquad \frac{\Gamma \vdash M : A \wedge B\,;\ \Delta}{\Gamma \vdash \mathbf{proj_2}\, M : B\,;\ \Delta}\ (\wedge_2\text{-E})$$

$$\frac{\Gamma \vdash M : A\,;\ \Delta}{\Gamma \vdash \mathbf{inj_1}\, M : A \vee B\,;\ \Delta}\ (\vee_1\text{-I}) \qquad \frac{\Gamma \vdash M : B\,;\ \Delta}{\Gamma \vdash \mathbf{inj_2}\, M : A \vee B\,;\ \Delta}\ (\vee_2\text{-I})$$

$$\frac{\Gamma \vdash L : A \vee B\,;\ \Delta \quad \Gamma \cup \{x : A\} \vdash M : C\,;\ \Delta \quad \Gamma \cup \{y : B\} \vdash N : C\,;\ \Delta}{\Gamma \vdash \mathbf{case}\ L\ x.M\ y.N : C\,;\ \Delta}\ (\vee\text{-E})$$

$$\frac{\Gamma \cup \{x : A\} \vdash M : B\,;\ \{\}}{\Gamma \vdash \lambda\, x.\, M : A \supset B\,;\ \Delta}\ (\supset\text{-I}) \qquad \frac{\Gamma \vdash M : A \supset B\,;\ \Delta \quad \Gamma \vdash N : A\,;\ \Delta}{\Gamma \vdash M\, N : B\,;\ \Delta}\ (\supset\text{-E})$$

$$\frac{\Gamma \vdash M : A\,;\ \Delta \cup \{u : E\}}{\Gamma \vdash \kappa\, u.\, M : A \triangleleft E\,;\ \Delta}\ (\triangleleft\text{-I}) \qquad \frac{\Gamma \vdash M : A \triangleleft E\,;\ \Delta}{\Gamma \vdash M\, u : A\,;\ \Delta \cup \{u : E\}}\ (\triangleleft\text{-E})$$

Example 2.2.5 (Derivations) Let Γ be as $\Gamma = \{x : A, f : A \supset A\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \cup \{y : B\} \vdash x : A\,;\ \{\}}\ (var)}{\Gamma \vdash \lambda\, y.\, x : B \supset A\,;\ \{u : A\}}\ (\supset\text{-I}) \quad \dfrac{\dfrac{\dfrac{}{\Gamma \vdash f : A \supset A\,;\ \{\}}\ (var) \quad \dfrac{}{\Gamma \vdash x : A\,;\ \{\}}\ (var)}{\Gamma \vdash f\, x : A\,;\ \{\}}\ (\supset\text{-E})}{\Gamma \vdash \mathbf{throw}\ u\ (f\ x) : B\,;\ \{u : A\}}\ (throw)}{\Gamma \vdash (\lambda\, y.\, x)\,(\mathbf{throw}\ u\ (f\ x)) : A\,;\ \{u : A\}}\ (\supset\text{-E})}{\Gamma \vdash \mathbf{catch}\ u\ ((\lambda\, y.\, x)\,(\mathbf{throw}\ u\ (f\ x))) : A\,;}\ (catch)}{\{x : A\} \vdash \lambda\, f.\, \mathbf{catch}\ u\ ((\lambda\, y.\, x)\,(\mathbf{throw}\ u\ (f\ x))) : (A \supset A) \supset A\,;}\ (\supset\text{-I})}{\vdash \lambda\, x.\, \lambda\, f.\, \mathbf{catch}\ u\ ((\lambda\, y.\, x)\,(\mathbf{throw}\ u\ (f\ x))) : A \supset (A \supset A) \supset A\,;}\ (\supset\text{-I})$$

There is nothing special except for $(throw)$, $(catch)$, $(\supset$-I$)$, $(\triangleleft$-I$)$ and $(\triangleleft$-E$)$. If we have a term M of a type E, we can treat the term **throw** $u\ M$ as if it belongs to an arbitrary type A, but in reality, the evaluation of **throw** $u\ M$ causes a throw-operation of the result of M to the tag u instead of returning a value of A. On the other hand, if we have a term M of A which may throw a value of A to the tag u during the evaluation of M, then we can treat **catch** $u\ M$ as a term of A which causes no throw-operation to u. Once we adopt the interpretation of typing judgements explained above, these rules for **catch** and **throw** are quite natural.

The most important point about our typing rules is a restriction on the rule $(\supset$-I$)$. We can introduce a λ-abstraction only if its body has no free tag variable, that is, the tag context of the premise must be empty. If we dropped this restriction, our intended interpretation of typing judgements would be affected. Consider the following example.

$$\dfrac{\dfrac{\dfrac{}{\{x:A\} \vdash x:A\,;\,\{\}}\,(var)}{\{x:A\} \vdash \mathbf{throw}\ u\ x : B\,;\,\{u:A\}}\,(throw)}{\{\} \vdash \lambda x.\,\mathbf{throw}\ u\ x : A \supset B\,;\,\{u:A\}}\,(\supset\text{-I})$$

The derived judgement says that the evaluation of $\lambda x.\,\mathbf{throw}\ u\ x$ terminates with a value of $A \supset B$, or a value of A is thrown to the tag u during the evaluation. But the evaluation of $\lambda x.\,\mathbf{throw}\ u\ x$ immediately terminates with itself, which is not a value of $A \supset B$. Note that $\mathbf{throw}\ u\ x$ is not evaluated until $\lambda x.\,\mathbf{throw}\ u\ x$ is applied to some value of the type A. From the logical point of view, this restriction on ($\supset$-I) is required to keep the system constructive. We shall discuss this point in Chapter 4.

The restriction on ($\supset$-I)-rule leads us to introduce the new connective $\triangleleft$. We can not construct any function that may throw something to the outside of the function without the new connective, because the body of the λ-abstraction must not have any tag variable. We can construct such a function by the new connective as follows. Let Γ be as $\Gamma = \{x : A \vee B\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash x : A \vee B\,;\,\{u:B\}}\,(var) \quad \begin{array}{c}\dfrac{}{\Gamma \cup \{y:A\} \vdash y:A\,;\,\{u:B\}}\,(var)\\ \vdots\end{array} \quad \dfrac{\dfrac{}{\Gamma \cup \{z:B\} \vdash z:B\,;\,\{\}}\,(var)}{\Gamma \cup \{z:B\} \vdash \mathbf{throw}\ u\ z : A\,;\,\{u:B\}}\,(throw)}{\Gamma \vdash \mathbf{case}\ x\ y.y\ z.(\mathbf{throw}\ u\ z) : A\,;\,\{u:B\}}\,(\vee\text{-E})}{\Gamma \vdash \kappa u.\,\mathbf{case}\ x\ y.y\ z.(\mathbf{throw}\ u\ z) : A \triangleleft B\,;\,\{\}}\,(\triangleleft\text{-I})}{\{\} \vdash \lambda x.\,\kappa u.\,\mathbf{case}\ x\ y.y\ z.(\mathbf{throw}\ u\ z) : (A \vee B) \supset (A \triangleleft B)\,;\,\{\}}\,(\supset\text{-I})$$

Let M be as $M = \lambda x.\,\kappa u.\,\mathbf{case}\ x\ y.y\ z.(\mathbf{throw}\ u\ z)$, and let N be a term of the type $A \vee B$. The function M can be used as follows.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \{\} \vdash M : (A \vee B) \supset (A \triangleleft B)\,;\,\{\}\end{array} \quad \begin{array}{c}\vdots\\ \{\} \vdash N : A \vee B\,;\,\{\}\end{array}}{\{\} \vdash M\,N : A \triangleleft B\,;\,\{\}}\,(\supset\text{-E})}{\{\} \vdash (M\,N)\,v : A\,;\,\{v:B\}}\,(\triangleleft\text{-E})$$

Normally the function M returns a value of A, otherwise it throws a value of B to the given tag v.

2.3 A realizability interpretation

In this section we give a realizability interpretation of $L^{\text{CBV}}_{c/t}$ to show how the catch/throw mechanism can be captured in it.

2.3.1 The realizability

Let $\mathcal{A}$ be a mapping which assigns a subset of *Const* to each atomic type. The realizability is defined relatively to this mapping $\mathcal{A}$.

Definition 2.3.1 (Realizability of types) Let V be a value, and A a type. We define a relation $\mathbf{r}$ between values and types as follows.

1. $V \mathbf{r} A$ iff $V \in \mathcal{A}(A)$, if A is an atomic type.
2. $V \mathbf{r} A_1 \wedge A_2$ iff $V = <V_1, V_2>$ for some V_1 and V_2 such that $V_1 \mathbf{r} A_1$ and $V_2 \mathbf{r} A_2$.
3. $V \mathbf{r} A_1 \vee A_2$ iff $V = \mathbf{inj}_i W$ and $W \mathbf{r} A_i$ for some W and i $(i = 1, 2)$.
4. $V \mathbf{r} A_1 \supset A_2$ iff $V = \lambda x. M$ for some term M such that for any value W, if $W \mathbf{r} A_1$, then $M[W/x] \underset{\text{CBV}}{\succ} V'$ and $V' \mathbf{r} A_2$ for some value V'.
5. $V \mathbf{r} A_1 \triangleleft A_2$ iff $V = \kappa u. M$ for some u and M such that
   (a) M is a value and $M \mathbf{r} A_1$, or
   (b) $M = \mathbf{throw}\ u\ W$ and $W \mathbf{r} A_2$ for some value W.

If the relation holds between a term and a type, we say that the term *realizes* the type, and the term is a *realizer* of the type.

Definition 2.3.2 (Interpretation) We define the interpretation of typing judgements as follows. The relation

$$\{x_1 : A_1, \ldots, x_m : A_m\} \models M : C\ ;\ \{u_1 : B_1, \ldots, u_n : B_n\}$$

holds if and only if for any closed terms $K_1, \ldots, K_m$ such that $K_i \underset{\text{CBV}}{\succ} W_i$ and $W_i \mathbf{r} A_i$ for some W_i $(1 \leq i \leq m)$,

1. $M[K_1/x_1, \ldots, K_m/x_m] \underset{\text{CBV}}{\succ} V$ and $V \mathbf{r} C$ for some V, or
2. $M[K_1/x_1, \ldots, K_m/x_m] \underset{\text{CBV}}{\succ} \mathcal{C}[\mathbf{throw}\ u_j\ V]$ and $V \mathbf{r} B_j$ for some j, V and $\mathcal{C}$ which does not capture u_j.

This interpretation is essentially the same as the standard realizability interpretation of NJ in the case that the type C does not include any occurrence of $\triangleleft$ and $n = 0$. It should also be noted that the logical connective $\triangleleft$ corresponds to the semicolon of a typing judgement as $\supset$ corresponds to $\vdash$.

Lemma 2.3.3 *Suppose that* $\{x_1 : A_1, \ldots, x_m : A_m\} \models M : C\ ;\ \{u_1 : B_1, \ldots, u_n : B_n\}$ *holds. and let* $y_1 \ldots y_m$ *and* $v_1 \ldots v_n$ *be a sequence of distinct individual variables and a sequence of distinct tag variables, respectively. Then*

$$\{y_1 : A_1, \ldots, y_m : A_m\} \models M[\vec{y}/\vec{x}, \vec{v}/\vec{u}]\ ;\ \{v_1 : B_1, \ldots, v_n : B_n\}$$

also holds, where $M[\vec{y}/\vec{x}, \vec{v}/\vec{u}]$ *stands for* $M[y_1/x_1, \ldots, y_m/x_m, v_1/u_1, \ldots, v_n/u_n]$.

Proof. Since $M[\vec{y}/\vec{x}, \vec{v}/\vec{u}][\vec{K}/\vec{y}] = M[\vec{K}/\vec{x}, \vec{v}/\vec{u}]$, straightforward from the definition of $\Gamma \models M : C\ ;\ \Delta$ by Proposition 2.1.16. $\square$

2.3.2 Soundness

The following soundness theorem assures us that we can regard the proofs of the formal system as programs which satisfy the specification defined by the realizability interpretation of the conclusion.

Theorem 2.3.4 (Soundness) *If* $\Gamma \vdash M : C\,;\,\Delta$ *is derivable in* $L_{c/t}^{\text{CBV}}$*, then* $\Gamma \models M : C\,;\,\Delta$ *holds.*

Proof. By induction on the structure of the derivation. Let Γ be as $\Gamma = \{x_1 : A_1, \ldots, x_m : A_m\}$, and let Δ be as $\Delta = \{u_1 : B_1, \ldots, u_n : B_n\}$. Let $K_1, \ldots, K_m$ be closed terms such that $K_i \underset{\text{CBV}}{\succ} W_i$ and $W_i \mathbf{\ r\ } A_i$ for some W_i $(1 \le i \le m)$. Each induction step is done by cases according to the rule applied in the last step of the derivation.

Case 1: *The last rule is* (*var*). Trivial.

Case 2: *The last rule is* (*let*). Let $\Gamma \vdash M_1 : D\,;\,\Delta$ and $\Gamma \cup \{z : D\} \vdash M_2 : C\,;\,\Delta$ be the premises, that is, $M = \mathbf{let}\ z = M_1.\ M_2$. By the induction hypothesis, $\Gamma \models M_1 : D\,;\,\Delta$ and $\Gamma \cup \{z : D\} \models M_2 : C\,;\,\Delta$ hold. Therefore for some V_1,

1. $M_1[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} V_1$ and $V_1 \mathbf{\ r\ } D$, or
2. $M_1[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ V_1]$ and $V_1 \mathbf{\ r\ } B_j$ for some j $(1 \le j \le n)$ and $\mathcal{C}'$ which does not capture u_j.

Let z' be a fresh individual variable. In the first case,

$$\begin{array}{rcl}
(\mathbf{let}\ z = M_1.\ M_2)[\vec{K}/\vec{x}] & = & \mathbf{let}\ z' = M_1[\vec{K}/\vec{x}].\ M_2[z'/z][\vec{K}/\vec{x}] \\
& \xrightarrow[\text{CBV}]{*} & \mathbf{let}\ z' = V_1.\ M_2[z'/z][\vec{K}/\vec{x}] \\
& \xrightarrow[\text{CBV}]{} & M_2[z'/z][\vec{K}/\vec{x}][V_1/z'] \\
& = & M_2[V_1/z][\vec{K}/\vec{x}].
\end{array}$$

Since $\Gamma \cup \{z : D\} \models M_2 : C\,;\,\Delta$ and $V_1 \mathbf{\ r\ } D$, for some V_2,

$$M_2[V_1/z][\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} V_2 \quad \text{and} \quad V_2 \mathbf{\ r\ } C, \quad \text{or}$$

$$M_2[V_1/z][\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}[\mathbf{throw}\ v_j\ V_2] \quad \text{and} \quad V_2 \mathbf{\ r\ } B_j$$

for some j $(1 \le j \le n)$ and $\mathcal{C}$ which does not capture v_j. Therefore, one of the two conditions of Definition 2.3.2 is satisfied. In the second case, let $\mathcal{C}$ be as $\mathcal{C} = \mathbf{let}\ z' = \mathcal{C}'.\ M_2[z'/z][\vec{K}/\vec{x}]$.

$$\begin{array}{rcl}
(\mathbf{let}\ z = M_1.\ M_2)[\vec{K}/\vec{x}] & = & \mathbf{let}\ z' = M_1[\vec{K}/\vec{x}].\ M_2[z'/z][\vec{K}/\vec{x}] \\
& \xrightarrow[\text{CBV}]{*} & \mathbf{let}\ z' = \mathcal{C}'[\mathbf{throw}\ u_j\ V_1].\ M_2[z'/z][\vec{K}/\vec{x}] \\
& = & \mathcal{C}[\mathbf{throw}\ u_j\ V_1].
\end{array}$$

Since $\mathcal{C}$ does not capture u_j, the second condition of Definition 2.3.2 is satisfied in this case. Therefore $\Gamma \models \mathbf{let}\ z = M_1.\ M_2\,;\,\Delta$ holds.

Case 3: *The last rule is (throw).* Let $\Gamma \vdash M' : B_k\,;\, \Delta'$ be the premise, where $\Delta = \Delta' \cup \{u_k : B_k\}$ and $M = \mathbf{throw}\ u_k\ M'$. By the induction hypothesis, for some V',

1. $M'[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} V'$ and $V'\ \mathbf{r}\ B_k$, or
2. $M[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ V']$ and $V'\ \mathbf{r}\ B_j$ for some j $(1 \le j \le n)$ and $\mathcal{C}'$ which does not capture u_j.

In the first case,

$$(\mathbf{throw}\ u_k\ M')[\vec{K}/\vec{x}] \;=\; \mathbf{throw}\ u_k\ M'[\vec{K}/\vec{x}] \;\xrightarrow[\text{CBV}]{*}\; \mathbf{throw}\ u_k\ V'.$$

Therefore the second condition of Definition 2.3.2 is satisfied since $V'\ \mathbf{r}\ B_k$. In the second case, let $\mathcal{C}$ be as $\mathcal{C} = \mathbf{throw}\ u_k\ \mathcal{C}'$.

$$\begin{array}{rcl} (\mathbf{throw}\ u_k\ M')[\vec{K}/\vec{x}] & = & \mathbf{throw}\ u_k\ M'[\vec{K}/\vec{x}] \\ & \xrightarrow[\text{CBV}]{*} & \mathbf{throw}\ u_k\ \mathcal{C}'[\mathbf{throw}\ u_j\ V'] \\ & = & \mathcal{C}[\mathbf{throw}\ u_j\ V']. \end{array}$$

Since $\mathcal{C}$ does not capture u_j, the second condition of Definition 2.3.2 is satisfied also in this case. Therefore $\Gamma \models \mathbf{throw}\ u_k\ M' : C\,;\, \Delta$ holds.

Case 4: *The last rule is (catch).* Let $\Gamma \vdash M' : C\,;\, \Delta \cup \{v : C\}$ be the premise, where $M = \mathbf{catch}\ v\ M'$. By the induction hypothesis, for some V',

1. $M'[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} V'$ and $V'\ \mathbf{r}\ C$,
2. $M'[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ v\ V']$ and $V'\ \mathbf{r}\ C$ for some $\mathcal{C}'$ which does not capture v, or
3. $M'[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ V']$, $V'\ \mathbf{r}\ B_j$ and $u_j \neq v$, for some j and $\mathcal{C}'$ which does not capture u_j.

In the first case,

$$(\mathbf{catch}\ v\ M')[\vec{K}/\vec{x}] \;=\; \mathbf{catch}\ v\ M'[\vec{K}/\vec{x}] \;\xrightarrow[\text{CBV}]{*}\; \mathbf{catch}\ v\ V' \;\xrightarrow[\text{CBV}]{}\; V'.$$

Therefore the first condition of Definition 2.3.2 is satisfied. In the second case, since $\mathcal{C}'$ does not capture v,

$$\begin{array}{rcl} (\mathbf{catch}\ v\ M')[\vec{K}/\vec{x}] & = & \mathbf{catch}\ v\ M'[\vec{K}/\vec{x}] \\ & \xrightarrow[\text{CBV}]{*} & \mathbf{catch}\ v\ \mathcal{C}'[\mathbf{throw}\ v\ V'] \\ = \text{ or } & \xrightarrow[\text{CBV}]{} & \mathbf{catch}\ v\ (\mathbf{throw}\ v\ V') \\ & \xrightarrow[\text{CBV}]{} & V'. \end{array}$$

That is, the first condition is satisfied also in this case. In the last case, let $\mathcal{C}$ be as $\mathcal{C} = \mathbf{catch}\ v\ \mathcal{C}'$.

$$\begin{array}{rcl} (\mathbf{catch}\ v\ M')[\vec{K}/\vec{x}] & = & \mathbf{catch}\ v\ M'[\vec{K}/\vec{x}] \\ & \xrightarrow[\text{CBV}]{*} & \mathbf{catch}\ v\ \mathcal{C}'[\mathbf{throw}\ u_j\ V'] \\ & = & \mathcal{C}[\mathbf{throw}\ u_j\ V']. \end{array}$$

Since $u_j \neq v$, $\mathcal{C}$ does not capture u_j. Therefore the second condition of Definition 2.3.2 is satisfied in this case. We now get $\Gamma \models \mathbf{catch}\ v\ M'\,;\, \Delta$.

Case 5: *The last rule is* ($\supset$-*I*). Let $\Gamma \cup \{z : C_1\} \vdash M' : C_2\,;\{\}$ be the premise, where $M = \lambda\, z.\, M'$ and $C = C_1 \supset C_2$. Let z' be a fresh individual variable.

$$(\lambda\, z.\, M')[\vec{K}/\vec{x}] = \lambda\, z'.\, M[z'/z][\vec{K}/\vec{x}] \in \mathit{Val}.$$

Assume that $W\ \mathbf{r}\ C_1$. By the induction hypothesis,

$$M[z'/z][\vec{K}/\vec{x}][W/z'] = M[W/z][\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} V' \quad \text{and} \quad V'\ \mathbf{r}\ C_2.$$

That is, $(\lambda\, z.\, M')[\vec{K}/\vec{x}]\ \mathbf{r}\ C_1 \supset C_2$.

Case 6: *The last rule is* ($\supset$-*E*). Let $\Gamma \vdash M_1 : D \supset C\,;\,\Delta$ and $\Gamma \vdash M_2 : D\,;\,\Delta$ be the premises, where $M = M_1\, M_2$. By the induction hypothesis on the first premise, for some V_1,

1. $M_1[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} V_1$ and $V_1\ \mathbf{r}\ D \supset C$, or
2. $M_1[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ V_1]$ and $V_1\ \mathbf{r}\ B_j$ for some j $(1 \le j \le n)$ and $\mathcal{C}'$ which does not capture u_j.

In the first case, since $V_1\ \mathbf{r}\ D \supset C$, we can assume that $V_1 = \lambda\, z.\, N$ for some z and N. Therefore,

$$(M_1\, M_2)[\vec{K}/\vec{x}] \quad = \quad M_1[\vec{K}/\vec{x}]\, M_2[\vec{K}/\vec{x}] \quad \xrightarrow[\text{CBV}]{*} \quad (\lambda\, z.\, N)\, M_2[\vec{K}/\vec{x}]$$

On the other hand, by induction hypothesis on the second premise, for some V_2,

$$M_2[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} V_2 \quad \text{and} \quad V_2\ \mathbf{r}\ C, \quad \text{or}$$

$$M_2[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}''[\mathbf{throw}\ u_j\ V_2] \quad \text{and} \quad V_2\ \mathbf{r}\ B_j$$

for some j $(1 \le j \le n)$ and $\mathcal{C}''$ which does not capture u_j. Therefore, $(M_1\, M_2)[\vec{K}/\vec{x}] \xrightarrow[\text{CBV}]{*} (\lambda\, z.\, N)\, V_2 \xrightarrow[\text{CBV}]{} N[V_2/z]$ or $(M_1\, M_2)[\vec{K}/\vec{x}] \xrightarrow[\text{CBV}]{*} (\lambda\, z.\, N)\, \mathcal{C}[\mathbf{throw}\ u_j\ V_2]$. Since $\lambda\, z.\, N\ \mathbf{r}\ D \supset C$ and $V_2\ \mathbf{r}\ D$, one of the two conditions of Definition 2.3.2 is satisfied. In the second case, let $\mathcal{C}$ be as $\mathcal{C} = \mathcal{C}'\, M_2[\vec{K}/\vec{x}]$.

$$(M_1\, M_2)[\vec{K}/\vec{x}] \quad \xrightarrow[\text{CBV}]{*} \quad \mathcal{C}'[\mathbf{throw}\ u_j\ V_1]\, M_2[\vec{K}/\vec{x}] \quad = \quad \mathcal{C}[\mathbf{throw}\ u_j\ V_1]$$

Since $\mathcal{C}$ does not capture u_j, the second condition of Definition 2.3.2 is satisfied in this case. Therefore, $\Gamma \models M_1\, M_2\,;\,\Delta$ holds.

Case 7: *The last rule is* ($\triangleleft$-*I*). Let $\Gamma \vdash M' : C_1\,;\,\Delta \cup \{v : C_2\}$ be the premise, where $M = \kappa\, v.\, M'$ and $C = C_1 \triangleleft C_2$.

$$(\kappa\, v.\, M')[\vec{K}/\vec{x}] = \kappa\, v.\, M'[\vec{K}/\vec{x}].$$

By the induction hypothesis, for some V',

1. $M'[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} V'$ and $V'\ \mathbf{r}\ C_1$,
2. $M'[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ v\ V']$ and $V'\ \mathbf{r}\ C_2$ for some $\mathcal{C}'$ which does not capture v, or
3. $M'[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ V']$, $V'\ \mathbf{r}\ B_j$ and $u_j \neq v$, for some j and $\mathcal{C}'$ which does not capture u_j.

In the first case,

$$(\kappa\, v.\, M')[\vec{K}/\vec{x}] \quad = \quad \kappa\, v.\, M'[\vec{K}/\vec{x}] \quad \xrightarrow[\text{CBV}]{*} \quad \kappa\, v.\, V',$$

and since V' $\mathbf{r}$ C_1, we get $\kappa\, v.\, V'$ $\mathbf{r}$ $C_1 \triangleleft C_2$. Therefore, the first condition of Definition 2.3.2 is satisfied. In the second case,

$$\begin{array}{rcl} (\kappa\, v.\, M')[\vec{K}/\vec{x}] & = & \kappa\, v.\, M'[\vec{K}/\vec{x}] \\ & \xrightarrow[\text{CBV}]{*} & \kappa\, v.\, \mathcal{C}'[\mathbf{throw}\ v\ V'] \\ & = \text{ or } \xrightarrow[\text{CBV}]{} & \kappa\, v.\, \mathbf{throw}\ v\ V'. \end{array}$$

Since V' $\mathbf{r}$ C_2, we get $\kappa\, v.\, \mathbf{throw}\ v\ V'$ $\mathbf{r}$ $C_1 \triangleleft C_2$. That is, the first condition is satisfied also in this case. In the third case, let $\mathcal{C}$ be as $\mathcal{C} = \kappa\, v.\, \mathcal{C}'$.

$$\begin{array}{rcl} (\kappa\, v.\, M')[\vec{K}/\vec{x}] & = & \kappa\, v.\, M'[\vec{K}/\vec{x}] \\ & \xrightarrow[\text{CBV}]{*} & \kappa\, v.\, \mathcal{C}'[\mathbf{throw}\ u_j\ V'] \\ & = & \mathcal{C}[\mathbf{throw}\ u_j\ V']. \end{array}$$

Since $u_j \neq v$, $\mathcal{C}$ does not capture u_j. Therefore the second condition is satisfied. We now get $\Gamma \models \kappa\, v.\, M' : C_1 \triangleleft C_2\, ;\, \Delta$.

Case 8: *The last rule is* $(\triangleleft\text{-}E)$. Let $\Gamma \vdash M' : C \triangleleft B_k\, ;\, \Delta'$ be the premise, where $M = M'\, u_k$ and $\Delta = \Delta' \cup \{u_k : B_k\}$.

$$(M'\, u_k)[\vec{K}/\vec{x}] = M'[\vec{K}/\vec{x}]\, u_k.$$

By the induction hypothesis, for some V',

1. $M'[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \kappa\, v.\, V'$ and V' $\mathbf{r}$ C for some v,
2. $M'[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \kappa\, v.\, \mathbf{throw}\ v\ V'$ and V' $\mathbf{r}$ B_k for some v, or
3. $M'[\vec{K}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ V']$ and V' $\mathbf{r}$ B_j for some j $(1 \leq j \leq n)$ and $\mathcal{C}'$ which does not capture u_j.

In the first case,

$$(M'\, u_k)[\vec{K}/\vec{x}] = M'[\vec{K}/\vec{x}]\, u_k \xrightarrow[\text{CBV}]{*} (\kappa\, v.\, V')\, u_k \xrightarrow[\text{CBV}]{} V'$$

That is, the first condition of Definition 2.3.2 is satisfied in this case. In the second case,

$$(M'\, u_k)[\vec{K}/\vec{x}] = M'[\vec{K}/\vec{x}]\, u_k \xrightarrow[\text{CBV}]{*} (\kappa\, v.\, \mathbf{throw}\ v\ V')\, u_k \xrightarrow[\text{CBV}]{} \mathbf{throw}\ u_k\ V'$$

Since V' $\mathbf{r}$ B_k, the second condition is satisfied. In the third case, Let $\mathcal{C}$ be as $\mathcal{C} = \mathcal{C}'\, u_k$.

$$(M'\, u_k)[\vec{K}/\vec{x}] = M'[\vec{K}/\vec{x}]\, u_k \xrightarrow[\text{CBV}]{*} \mathcal{C}'[\mathbf{throw}\ u_j\ V']\, u_k = \mathcal{C}[\mathbf{throw}\ u_j\ V']$$

Since $\mathcal{C}$ does not capture u_j, the second condition is satisfied in this case. We now get $\Gamma \models M'\, u_k : C\, ;\, \Delta$.

Case 10: *The last rule is one of the others.* We get $\Gamma \models M : C\, ;\, \Delta$ similarly. □

Corollary 2.3.5 *If* $\{\} \vdash M : C\, ;\, \{\}$ *is derivable in* $L_{c/t}^{\text{CBV}}$, *then* $M \underset{\text{CBV}}{\succ} V$ *for some value* V.

Proof. Straightforward from Theorem 2.3.4. □

Chapter 3

The conventional implementation

In this section we imitate the standard implementation of the catch/throw mechanism by an abstract stack machine, and give a realizability interpretation of the formal system in terms of the abstract machine which is equivalent to the realizability defined in Section 2.3.

3.1 Definition of the machine

We design the machine only to illustrate how the catch/throw mechanism works. Other mechanisms required for the evaluation of terms remain abstract (cf. [17, 30]). First, we extend the syntax. The extended part is used by the machine internally.

Definition 3.1.1 (Tag constants) We assume that a set *Tconst* which is a representation of the set of natural numbers is given. Elements of *Tconst* are called *tag constants.* We use $\overline{n}$ to denote the tag constant that represents a natural number n.

Definition 3.1.2 (Tag^+) We define a set Tag^+ by

$$Tag^+ ::= Tconst \quad | \quad Tvar\ .$$

Elements of Tag^+ are called *internal tags.* We use $t, t', \ldots$ to denote internal tags.

Definition 3.1.3 (Val^+ and $Term^+$) We define two sets Val^+ and $Term^+$, simultaneously, as follows.

$$\begin{array}{rcl}
Val^+ & ::= & Const \quad | \quad \lambda\, Var.\, Term^+ \quad | \quad <Val^+,\ Val^+> \\
& | & \mathbf{inj_1}\, Val^+ \quad | \quad \mathbf{inj_2}\, Val^+ \\
& | & \kappa\, Tvar.\, \mathbf{val}\ Val^+ \quad | \quad \kappa\, Tvar.\, \mathbf{throw}\ Tvar\ (\mathbf{val}\ Val^+)
\end{array}$$

$$\begin{array}{rcl}
Term^+ & ::= & Const \quad | \quad Var \quad | \quad \mathbf{val}\ Val^+ \quad | \quad \mathbf{let}\ Var{=}Term^+.\, Term^+ \\
& | & \mathbf{throw}\ Tag^+\ Term^+ \quad | \quad \mathbf{catch}\ Tvar\ Term^+ \\
& | & \lambda\, Var.\, Term^+ \quad | \quad Term^+\, Term^+ \\
& | & <Term^+,\ Term^+> \quad | \quad \mathbf{proj_1}\, Term^+ \quad | \quad \mathbf{proj_2}\, Term^+ \\
& | & \mathbf{inj_1}\, Term^+ \quad | \quad \mathbf{inj_2}\, Term^+ \quad | \quad \mathbf{case}\ Term^+\ \ Var.\, Term^+\ \ Var.\, Term^+ \\
& | & \kappa\ Tvar.\ Term^+ \quad | \quad Term^+\, Tag^+\ ,
\end{array}$$

where every element of Val^+ must be closed and must not have any occurrence of tag constants. Elements of Val^+ are called *internal values*. We use Q, R, ... to denote internal values. Elements of $Term^+$ are called *internal terms*. We use $e, f, g, \ldots$ to denote internal terms. Note that if $\lambda\, x.\, e$ is an internal value, then $FIV(e) \subset \{x\}$ and $FTV(e) = \{\}$ since every internal value must be closed.

Proposition 3.1.4 *$Val^+ \subset Term^+$ and $Term \subset Term^+$.*

Proof. Obvious from the definition. □

We now have new terms such as $\mathbf{val}\ (\lambda\, x.\, e)$, $\mathbf{throw}\ \overline{3}\ e$ and $e\,\overline{0}$, as internal terms as well as ordinary terms.

Definition 3.1.5 Let e be an internal term which does not have any occurrence of tag constants. We assign a term $\overline{e}$ to e as follows.

$$
\begin{array}{rclrcl}
\overline{c} & = & c & \overline{x} & = & x \\
\overline{\mathbf{val}\ e} & = & \overline{e} & \overline{\mathbf{let}\ x{=}e_1.\ e_2} & = & \mathbf{let}\ x{=}\overline{e}_1.\ \overline{e}_2 \\
\overline{\mathbf{throw}\ u\ e} & = & \mathbf{throw}\ u\ \overline{e} & \overline{\mathbf{catch}\ u\ e} & = & \mathbf{catch}\ u\ \overline{e} \\
\overline{\lambda\, x.\, e} & = & \lambda\, x.\, \overline{e} & \overline{e_1\, e_2} & = & \overline{e}_1\, \overline{e}_2 \\
\overline{<e_1,\ e_2>} & = & <\overline{e}_1,\ \overline{e}_2> & \overline{\mathbf{proj}_i\ e} & = & \mathbf{proj}_i\ \overline{e} \\
\overline{\mathbf{inj}_i\ e} & = & \mathbf{inj}_i\ \overline{e} & \overline{\mathbf{case}\ e_0\ x.e_1\ y.e_2} & = & \mathbf{case}\ \overline{e}_0\ x.\overline{e}_1\ y.\overline{e}_2 \\
\overline{\kappa\, u.\, e} & = & \kappa\, u.\, \overline{e} & \overline{e\, u} & = & \overline{e}\, u
\end{array}
$$

That is, $\overline{e}$ is the term obtained from e by stripping all **val**'s occurring in e.

Proposition 3.1.6 *Let e be a closed internal term. If e has no occurrence of tag constants, then $\overline{e}$ is a closed term.*

Proof. Straightforward induction on the structure of e. □

Proposition 3.1.7 *$\overline{Q} \in Val$ for any internal value Q.*

Proof. Straightforward induction on the structure of Q. Note that any internal value has no occurrence of tag constants. □

Proposition 3.1.8 *Let M be a term, $x_1, \ldots, x_n$ individual variables, and $Q_1, \ldots, Q_n$ internal values.*

$$\overline{M[\mathbf{val}\ Q_1/x_1, \ldots, \mathbf{val}\ Q_n/x_n]} = M[\overline{Q}_1/x_1, \ldots, \overline{Q}_n/x_n].$$

Proof. Straightforward induction on the structure of M. □

Definition 3.1.9 (*Seg*) We define a set Seg by

$$
\begin{array}{rcl}
Seg & ::= & \mathbf{let}\ Var{=}{*}.\ Term^+ \ \mid\ \mathbf{throw}\ Tag^+ * \ \mid\ \mathbf{catch}\ Tvar * \\
& \mid & * \, Term^+ \ \mid\ Val^+ * \\
& \mid & <*,\ Term^+> \ \mid\ <Val^+,\ *> \ \mid\ \mathbf{proj_1} * \ \mid\ \mathbf{proj_2} * \\
& \mid & \mathbf{inj_1} * \ \mid\ \mathbf{inj_2} * \ \mid\ \mathbf{case} * \ Var.Term^+\ Var.Term^+ \\
& \mid & \kappa\, Tvar.\, * \ \mid\ \kappa\, Tvar.\, \mathbf{throw}\ Tvar * \ \mid\ * \, Tag^+\ ,
\end{array}
$$

where $\kappa\, Tvar.\, \mathbf{throw}\ Tvar *$ must be closed, that is, the two tag variables are identical. Elements of Seg are called *context segments*. We use $\mathcal{S}$, $\mathcal{S}'$, ... to denote context segments.

The abstract machine has a stack. The state of the machine is determined only by this stack state. We represent a stack state as follows.

$$\begin{array}{l} \text{bottom} \leftarrow \qquad\qquad \rightarrow \text{top} \\ [\, \mathcal{S}_1,\ \mathcal{S}_2,\ \mathcal{S}_3,\ \ldots,\ \mathcal{S}_n,\ e\,], \end{array}$$

where $\mathcal{S}_1, \ldots, \mathcal{S}_n$ are context segments and e is a closed internal term.

Definition 3.1.10 A stack state $[\ \mathcal{S}_1,\ \mathcal{S}_2,\ \mathcal{S}_3,\ \ldots,\ \mathcal{S}_n,\ e]$ is *valid* if $\mathcal{S}_1 \ldots \mathcal{S}_n$ are context segments and e is a closed internal term.

Definition 3.1.11 (Transition rules) The abstract machine changes its state according to the following table, where $\vec{\mathcal{S}}$ denotes a sequence of context segments.

$$\begin{array}{rcl}
[\,\vec{\mathcal{S}},\ c\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{val}\ c] \\
[\,\vec{\mathcal{S}},\ \mathbf{let}\ x{=}e_1.\ e_2\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{let}\ x{=}{*}.\ e_2,\ e_1\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{let}\ x{=}{*}.\ e_2,\ \mathbf{val}\ e_1\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ e_2[\mathbf{val}\ e_1/x]\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{throw}\ t\ e\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{throw}\ t\ {*},\ e\,] \\
[\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_n,\ \mathcal{S}_{n+1},\ \ldots,\ \mathcal{S}_{n+m}, & & \\
\mathbf{throw}\ \overline{n}\ {*},\ \mathbf{val}\ e\,] & \Rightarrow & [\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_n,\ \mathbf{val}\ e\,] \\
[\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_n,\ \mathbf{catch}\ u\ e\,] & \Rightarrow & [\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_n,\ e[\overline{n}/u]\,] \\
[\,\vec{\mathcal{S}},\ \lambda\, x.\, e\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\lambda\, x.\, e)\,] \\
[\,\vec{\mathcal{S}},\ e_1\, e_2\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ {*}\, e_2,\ e_1\,] \\
[\,\vec{\mathcal{S}},\ {*}\, e_2,\ \mathbf{val}\ e_1\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ e_1\, {*},\ e_2\,] \\
[\,\vec{\mathcal{S}},\ (\lambda\, x.\, e_1)\, {*},\ \mathbf{val}\ e_2\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ e_1[\mathbf{val}\ e_2/x]\,] \\
[\,\vec{\mathcal{S}},\ {<}e_1,\ e_2{>}\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ {<}{*},\ e_2{>},\ e_1\,] \\
[\,\vec{\mathcal{S}},\ {<}{*},\ e_2{>},\ \mathbf{val}\ e_1\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ {<}e_1,\ {*}{>},\ e_2\,] \\
[\,\vec{\mathcal{S}},\ {<}e_1,\ {*}{>},\ \mathbf{val}\ e_2\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{val}\ {<}e_1,\ e_2{>}\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{proj_1}\, e\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{proj_1}\, {*},\ e\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{proj_1}\, {*},\ \mathbf{val}\ {<}e_1,\ e_2{>}\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{val}\ e_1\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{proj_2}\, e\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{proj_2}\, {*},\ e\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{proj_2}\, {*},\ \mathbf{val}\ {<}e_1,\ e_2{>}\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{val}\ e_2\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{inj_1}\, e\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{inj_1}\, {*},\ e\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{inj_1}\, {*},\ \mathbf{val}\ e\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\mathbf{inj_1}\, e)\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{inj_2}\, e\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{inj_2}\, {*},\ e\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{inj_2}\, {*},\ \mathbf{val}\ e\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\mathbf{inj_2}\, e)\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{case}\ e_0\ x.e_1\ y.e_2\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{case}\ {*}\ x.e_1\ y.e_2,\ e_0\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{case}\ {*}\ x.e_1\ y.e_2,\ \mathbf{val}\ (\mathbf{inj_1}\, e_0)\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ e_1[\mathbf{val}\ e_0/x]\,] \\
[\,\vec{\mathcal{S}},\ \mathbf{case}\ {*}\ x.e_1\ y.e_2,\ \mathbf{val}\ (\mathbf{inj_2}\, e_0)\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ e_2[\mathbf{val}\ e_0/y]\,] \\
[\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_n,\ \kappa\, u.\, e\,] & \Rightarrow & [\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_n,\ \kappa\, u.\, \mathbf{throw}\ u\ {*}, \\
& & \quad \mathbf{throw}\ \overline{n}\ {*},\ \kappa\, u.\, {*},\ e[\overline{n{+}1}/u]\,]
\end{array}$$

$$\begin{array}{rcl}
[\,\vec{\mathcal{S}},\ \kappa\, u.\, *,\ \mathbf{val}\ e\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\kappa\, u.\, \mathbf{val}\ e)\,] \\
[\,\vec{\mathcal{S}},\ \kappa\, u.\, \mathbf{throw}\ u\ *,\ \mathbf{val}\ e\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\kappa\, u.\, \mathbf{throw}\ u\ (\mathbf{val}\ e))\,] \\
[\,\vec{\mathcal{S}},\ e\, t\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ *t,\ e\,] \\
[\,\vec{\mathcal{S}},\ *t,\ \mathbf{val}\ (\kappa\, u.\, e)\,] & \Rightarrow & [\,\vec{\mathcal{S}},\ e[t/u]\,]
\end{array}$$

Let $\stackrel{*}{\Rightarrow}$ be the transitive and reflexive closure of the relation $\Rightarrow$.

Example 3.1.12

$$\begin{array}{rl}
\multicolumn{2}{l}{[\,(\mathbf{catch}\ u\ ((\lambda\, z.\, e)\,(\mathbf{throw}\ u\ (\lambda\, x.\, x))))\, c\,]} \\
\Rightarrow & [\,*c,\ \mathbf{catch}\ u\ ((\lambda\, z.\, e)\,(\mathbf{throw}\ u\ (\lambda\, x.\, x)))\,] \\
\Rightarrow & [\,*c,\ (\lambda\, z.\, e)\,(\mathbf{throw}\ \overline{1}\ (\lambda\, x.\, x))\,] \\
\Rightarrow & [\,*c,\ *(\mathbf{throw}\ \overline{1}\ (\lambda\, x.\, x)),\ \lambda\, z.\, e\,] \\
\Rightarrow & [\,*c,\ *(\mathbf{throw}\ \overline{1}\ (\lambda\, x.\, x)),\ \mathbf{val}\ (\lambda\, z.\, e)\,] \\
\Rightarrow & [\,*c,\ (\lambda\, z.\, e)\,*,\ \mathbf{throw}\ \overline{1}\ (\lambda\, x.\, x)\,] \\
\Rightarrow & [\,*c,\ (\lambda\, z.\, e)\,*,\ \mathbf{throw}\ \overline{1}\ *,\ \lambda\, x.\, x\,] \\
\Rightarrow & [\,*c,\ (\lambda\, z.\, e)\,*,\ \mathbf{throw}\ \overline{1}\ *,\ \mathbf{val}\ (\lambda\, x.\, x)\,] \\
\Rightarrow & [\,*c,\ \mathbf{val}\ (\lambda\, x.\, x)\,] \\
\Rightarrow & [\,(\lambda\, x.\, x)\,*,\ c\,] \\
\Rightarrow & [\,(\lambda\, x.\, x)\,*,\ \mathbf{val}\ c\,] \\
\Rightarrow & [\,\mathbf{val}\ c\,]
\end{array}$$

Note that when the stack is in a state $[\ \mathcal{S}_1,\ \mathcal{S}_2,\ \ldots,\ \mathcal{S}_{n-1},\ \mathcal{S}_n,\ e\ ]$, the composition $\mathcal{S}_1[\mathcal{S}_2[\ldots[\mathcal{S}_{n-1}[\mathcal{S}_n]]\ldots]]$ of the context segments represents the evaluation contexts of the internal term e, i.e., the continuation after the evaluation of e. The catch/throw mechanism of the machine provides a restricted access to the continuation through tags. Observe that we need not any explicit copying of evaluation contexts to provide the mechanism.

Proposition 3.1.13 *Let* $[\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_m,\ e\,]$ *be a valid stack state. If*

$$[\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_m,\ e\,] \Rightarrow [\,\mathcal{S}'_1,\ \ldots,\ \mathcal{S}'_n,\ e'\,],$$

then $[\ \mathcal{S}'_1,\ \ldots,\ \mathcal{S}'_n,\ e'\ ]$ *is also valid.*

Proof. Obvious from the definition of transition rules. Note that e is closed since the state is valid. □

Definition 3.1.14 Let $\vec{\mathcal{S}}$ and $\vec{\mathcal{S}'}$ be sequences of context segments, and let e and e' be internal terms. $[\ \vec{\mathcal{S}'},\ e'\ ]$ is a *final state* of $[\ \vec{\mathcal{S}},\ e\ ]$, if $[\ \vec{\mathcal{S}},\ e\ ] \stackrel{*}{\Rightarrow} [\ \vec{\mathcal{S}'},\ e'\ ]$ and no transition rule is applicable to $[\,\vec{\mathcal{S}'},\ e'\,]$.

The rule applicable for a state is unique by the definition of the transition rules. Therefore, if $[e] \stackrel{*}{\Rightarrow} [\mathbf{val}\ Q]$, the internal value Q is unique.

3.2 Validity of the machine

In this subsection we discuss the validity of the abstract machine relative to the semantics given in Section 2.1.2. We show that

1. $M \underset{\text{CBV}}{\succ} V$ implies $[\, M \,] \overset{*}{\Rightarrow} [\, \textbf{val}\ Q \,]$ for some Q such that $\overline{Q} = V$, and
2. $[\, M \,] \overset{*}{\Rightarrow} [\, \textbf{val}\ Q \,]$ implies $M \underset{\text{CBV}}{\succ} \overline{Q}$.

for any closed term M.

Definition 3.2.1 Let e and e' be internal terms, and let $\mathcal{S}_1, \ldots, \mathcal{S}_m$ and $\mathcal{S}'_1, \ldots, \mathcal{S}'_n$ be sequences of context segments. Let l be a natural number. We define a relation $\underset{l}{\Rightarrow}$ by

$$[\, \mathcal{S}_1, \ldots, \mathcal{S}_m,\, e \,] \underset{l}{\Rightarrow} [\, \mathcal{S}'_1, \ldots, \mathcal{S}'_n,\, e' \,] \ \text{ iff } \begin{cases} [\, \mathcal{S}_1, \ldots, \mathcal{S}_m,\, e \,] \Rightarrow [\, \mathcal{S}'_1, \ldots, \mathcal{S}'_n,\, e' \,], \\ l \le m,\, n, \ \text{ and} \\ \mathcal{S}_i = \mathcal{S}'_i \quad (1 \le i \le l). \end{cases}$$

Let $\overset{*}{\underset{l}{\Rightarrow}}$ be the transitive and reflexive closure of the relation $\underset{l}{\Rightarrow}$.

The relation $\Rightarrow$ is equivalent to $\underset{0}{\Rightarrow}$, and $\underset{n}{\Rightarrow}$ implies $\underset{m}{\Rightarrow}$ if $m < n$. Note that if $[\, \mathcal{S}_1, \ldots, \mathcal{S}_m,\, e \,] \underset{l}{\Rightarrow} [\, \mathcal{S}'_1, \ldots, \mathcal{S}'_n,\, e' \,]$, then this transition does not depend on $\mathcal{S}_1, \ldots, \mathcal{S}_l$.

Definition 3.2.2 Let M be a term such that $M \underset{\text{CBV}}{\succ} N$ for some N. The rewriting path from M to N is unique. We denote the length of the path by $len(M)$.

Definition 3.2.3 Let M be a term. We define $|M|$ by

$$\begin{array}{rcl} |c| \ = \ |x| & = & 1 \\ |\textbf{let}\ x{=}M.\, N| & = & 1 + max(|M|, |N|) \\ |\textbf{throw}\ u\ M| \ = \ |\textbf{catch}\ u\ M| \ = \ |\lambda\, x.\, M| & = & 1 + |M| \\ |M\, N| \ = \ |{<}M,\ N{>}| & = & 1 + max(|M|, |N|) \\ |\textbf{proj}_i\ M| \ = \ |\textbf{inj}_i\ M| & = & 1 + |M| \\ |\textbf{case}\ L\ x.M\ y.N| & = & 1 + max(|L|, |M|, |N|) \\ |\kappa\, x.\, M| \ = \ |M\, u| & = & 1 + |M|. \end{array}$$

Theorem 3.2.4 *Let $\vec{\mathcal{S}}$ be a sequence of context segments whose lengths is l. Let $x_1, \ldots, x_m$ be individual variables, and $R_1, \ldots, R_m$ internal values. Let $u_1, \ldots, u_n$ be tag variables, and $\overline{k}_1, \ldots, \overline{k}_n$ tag constants such that $k_i \le l$ for any i. Let M be a term such that $FIV(M) \subset \{x_1, \ldots, x_m\}$ and $FTV(M) \subset \{u_1, \ldots, u_n\}$. Let $M[\overline{\vec{R}}/\vec{x}]$ and $M[\textbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]$ be abbreviations for $M[\overline{R}_1/x_1, \ldots, \overline{R}_m/x_m]$ and $M[\textbf{val}\ R_1/x_1, \ldots, \textbf{val}\ R_m/x_m, \overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]$, respectively.*

1. *If $M[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} V$, then*

$$[\, \vec{\mathcal{S}},\ M[\textbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}] \,] \overset{*}{\underset{l}{\Rightarrow}} [\, \vec{\mathcal{S}},\ \textbf{val}\ Q \,],$$

   for some internal value Q such that $\overline{Q} = V$.

2. *If $M[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}[\textbf{throw}\ u_j\ V]$ for some j and $\mathcal{C}$ which does not capture u_j, then*

$$[\, \vec{\mathcal{S}},\ M[\textbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}] \,] \overset{*}{\underset{l}{\Rightarrow}} [\, \vec{\mathcal{S}},\ \vec{\mathcal{S}}',\ \textbf{throw}\ \overline{k}_j\ {*},\ \textbf{val}\ Q \,],$$

   for some sequence of context segments $\vec{\mathcal{S}}'$ and some internal value Q such that $\overline{Q} = V$.

Proof. Induction on the lexicographic ordering of $len(M[\overline{\vec{R}}/\vec{x}])$ and $|M[\overline{\vec{R}}/\vec{x}]|$.

Case 1: *M is an individual constant.* In this case $M[\overline{\vec{R}}/\vec{x}] = M$. Therefore, we get $V = M$ if $M[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} V$. By the definition of transition rules,

$$[\ \vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ] = [\ \vec{\mathcal{S}},\ M\ ] \underset{l}{\Rightarrow} [\ \vec{\mathcal{S}},\ \mathbf{val}\ M\ ].$$

Let Q be as $Q = M$. Then we get $\overline{Q} = V$ since M is an individual constant. Note that $M[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\not\succ} \mathcal{C}[\mathbf{throw}\ u_i\ V]$.

Case 2: *M is an individual variable.* We get $M[\overline{\vec{R}}/\vec{x}] = M$ or $M[\overline{\vec{R}}/\vec{x}] = \overline{R}_i$ for some i. Therefore $M[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\not\succ} \mathcal{C}[\mathbf{throw}\ u_j\ V]$. Suppose that $M[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} V$, i.e., $M = x_i$ and $\overline{R}_i \underset{\mathrm{CBV}}{\succ} V$. Since $\overline{R}_i$ is a value, we get $\overline{R}_i = V$. Let Q be as $Q = R_i$. Obviously $[\ \vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ] \overset{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \mathbf{val}\ R_i\ ]$ since $M[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}] = \mathbf{val}\ R_i$.

Case 3: $M = \mathbf{let}\ y{=}M_1.\ M_2$ *for some y, M_1 and M_2.* We can assume $y \neq x_i$ for any i. First, suppose that $M[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} V$. By the definition of rewriting rules, $M_1[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} V_1$ and $M_2[\overline{\vec{R}}/\vec{x}][V_1/y] \underset{\mathrm{CBV}}{\succ} V$ for some value V_1. Note that $len(M_1[\overline{\vec{R}}/\vec{x}])$, $len(M_2[\overline{\vec{R}}/\vec{x}][V_1/y]) < len(M[\overline{\vec{R}}/\vec{x}])$. Therefore, for some internal values Q_1 and Q such that $\overline{Q}_1 = V_1$ and $\overline{Q} = V$,

$$\begin{array}{lll}
\multicolumn{2}{l}{[\ \vec{\mathcal{S}},\ (\mathbf{let}\ y{=}M_1.\ M_2)[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ]} & \\
\underset{l}{\Rightarrow} & [\ \vec{\mathcal{S}},\ \mathbf{let}\ y{=}*.\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}],\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ] & \\
\overset{*}{\underset{l}{\Rightarrow}} & [\ \vec{\mathcal{S}},\ \mathbf{let}\ y{=}*.\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}],\ \mathbf{val}\ Q_1\ ] & \text{(by ind. hyp.)} \\
\underset{l}{\Rightarrow} & [\ \vec{\mathcal{S}},\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \mathbf{val}\ Q_1/y, \vec{\overline{k}}/\vec{u}]\ ] & \\
\overset{*}{\underset{l}{\Rightarrow}} & [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ] & \text{(by ind. hyp.).}
\end{array}$$

Next suppose that $M[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} \mathcal{C}[\mathbf{throw}\ u_j\ V]$. By the definition of rewriting rules,

1. $M_1[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ V]$ and $\mathcal{C} = \mathbf{let}\ y{=}\mathcal{C}'.\ M_2$ for some $\mathcal{C}'$, or
2. $M_1[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} V_1$ and $M_2[\overline{\vec{R}}/\vec{x}][V_1/x] \underset{\mathrm{CBV}}{\succ} \mathcal{C}[\mathbf{throw}\ u_j\ V]$ for some value V_1.

In the first case, $len(M_1[\overline{\vec{R}}/\vec{x}]) = len(M[\overline{\vec{R}}/\vec{x}])$ and $|M_1[\overline{\vec{R}}/\vec{x}]| < |M[\overline{\vec{R}}/\vec{x}]|$. Therefore by the induction hypothesis, for some $\vec{\mathcal{S}}'$ and Q such that $\overline{Q} = V$,

$$\begin{array}{ll}
\multicolumn{2}{l}{[\ \vec{\mathcal{S}},\ (\mathbf{let}\ y{=}M_1.\ M_2)[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ]} \\
\underset{l}{\Rightarrow} & [\ \vec{\mathcal{S}},\ \mathbf{let}\ y{=}*.\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}],\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ] \\
\overset{*}{\underset{l}{\Rightarrow}} & [\ \vec{\mathcal{S}},\ \mathbf{let}\ y{=}*.\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}],\ \vec{\mathcal{S}}',\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q\ ].
\end{array}$$

In the second case, $len(M_1[\overline{\vec{R}}/\vec{x}])$, $len(M_2[\overline{\vec{R}}/\vec{x}][V_1/x]) < len(M[\overline{\vec{R}}/\vec{y}])$. Therefore, for some $\vec{\mathcal{S}}'$, Q_1 and Q such that $\overline{Q}_1 = V_1$ and $\overline{Q} = V$,

$$\begin{array}{lll}
\multicolumn{2}{l}{[\ \vec{\mathcal{S}},\ (\mathbf{let}\ y{=}M_1.\ M_2)[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ]} & \\
\underset{l}{\Rightarrow} & [\ \vec{\mathcal{S}},\ \mathbf{let}\ y{=}*.\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}],\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ] & \\
\overset{*}{\underset{l}{\Rightarrow}} & [\ \vec{\mathcal{S}},\ \mathbf{let}\ y{=}*.\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}],\ \mathbf{val}\ Q_1\ ] & \text{(by ind. hyp.)} \\
\underset{l}{\Rightarrow} & [\ \vec{\mathcal{S}},\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \mathbf{val}\ Q_1/y, \vec{\overline{k}}/\vec{u}]\ ] & \\
\overset{*}{\underset{l}{\Rightarrow}} & [\ \vec{\mathcal{S}},\ \vec{\mathcal{S}}',\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q\ ] & \text{(by ind. hyp.).}
\end{array}$$

Case 4: M = **throw** $v\ M'$ *for some* v *and* M'. Since $M[\vec{\vec{R}}/\vec{x}] \not\succ_{\text{CBV}} V$, suppose that $M \succ_{\text{CBV}} \mathcal{C}[\textbf{throw}\ u_j\ V]$. By the definition of rewriting rules,

1. $M'[\vec{\vec{R}}/\vec{x}] \succ_{\text{CBV}} \mathcal{C}'[\textbf{throw}\ u_j\ V]$ and $\mathcal{C} = \textbf{throw}\ v\ \mathcal{C}'$ for some $\mathcal{C}'$, or
2. $M'[\vec{\vec{R}}/\vec{x}] \succ_{\text{CBV}} V$ and $v = u_j$.

In the first case, $len(M'[\vec{\vec{R}}/\vec{x}]) = len(M[\vec{\vec{R}}/\vec{x}])$ and $|M'[\vec{\vec{R}}/\vec{x}]| < |M[\vec{\vec{R}}/\vec{x}]|$. Therefore by the induction hypothesis, for some $\vec{\mathcal{S}}'$ and Q such that $\overline{Q} = V$,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ (\textbf{throw}\ v\ M')[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}]\,] & = & [\,\vec{\mathcal{S}},\ \textbf{throw}\ v[\vec{\vec{k}}/\vec{u}]\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}]\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \textbf{throw}\ v[\vec{\vec{k}}/\vec{u}]\ *,\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}]\,] \\ & \overset{*}{\underset{l}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \textbf{throw}\ v[\vec{\vec{k}}/\vec{u}]\ *,\ \vec{\mathcal{S}}',\ \textbf{throw}\ \overline{k}_j\ *,\ \textbf{val}\ Q]. \end{array}$$

Similarly in the second case, for some Q such that $\overline{Q} = V$,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ (\textbf{throw}\ v\ M')[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}]\,] & = & [\,\vec{\mathcal{S}},\ \textbf{throw}\ \overline{k}_j\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}]\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \textbf{throw}\ \overline{k}_j\ *,\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}]\,] \\ & \overset{*}{\underset{l}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \textbf{throw}\ \overline{k}_j\ *,\ \textbf{val}\ Q\,], \end{array}$$

by the induction hypothesis.

Case 5: M = **catch** $v\ M'$ *for some* v *and* M'. We can assume that $v \neq u_i$ for any i. First, suppose that $M[\vec{\vec{R}}/\vec{x}] \succ_{\text{CBV}} V$. By the definition of rewriting rules,

1. $M'[\vec{\vec{R}}/\vec{x}] \succ_{\text{CBV}} V$, or
2. $M'[\vec{\vec{R}}/\vec{x}] \succ_{\text{CBV}} \mathcal{C}'[\textbf{throw}\ v\ V]$ for some $\mathcal{C}'$ which does not capture v.

In the first case, since $len(M'[\vec{\vec{R}}/\vec{x}]) < len(M[\vec{\vec{R}}/\vec{x}])$, by the induction hypothesis, for some Q such that $\overline{Q} = V$,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ (\textbf{catch}\ v\ M')[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}]\,] & = & [\,\vec{\mathcal{S}},\ \textbf{catch}\ v\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}]\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}, \overline{l}/v]\,] \\ & \overset{*}{\underset{l}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \textbf{val}\ Q\,]. \end{array}$$

Similarly in the second case, since $len(M'[\vec{\vec{R}}/\vec{x}]) < len(M[\vec{\vec{R}}/\vec{x}])$,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ (\textbf{catch}\ v\ M')[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}]\,] & = & [\,\vec{\mathcal{S}},\ \textbf{catch}\ v\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}]\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\vec{k}}/\vec{u}, \overline{l}/v]\,] \\ & \overset{*}{\underset{l}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}}',\ \textbf{throw}\ \overline{l}\ *,\ \textbf{val}\ Q\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \textbf{val}\ Q\,] \end{array}$$

for some $\vec{\mathcal{S}}'$ and Q such that $\overline{Q} = V$.

Next suppose that $M[\vec{\vec{R}}/\vec{x}] \succ_{\text{CBV}} \mathcal{C}[\textbf{throw}\ u_j\ V]$. Since $\mathcal{C}[\textbf{throw}\ u_j\ V]$ is not a value, $M'[\vec{\vec{R}}/\vec{x}] \succ_{\text{CBV}} \mathcal{C}'[\textbf{throw}\ u_j\ V]$ and $\mathcal{C}$ = **catch** $v\ \mathcal{C}'$ for some $\mathcal{C}'$. Since $len(M'[\vec{\vec{R}}/\vec{x}]) =$

$len(M[\vec{\overline{R}}/\vec{x}])$ and $|M'[\vec{\overline{R}}/\vec{x}]| < |M[\vec{\overline{R}}/\vec{x}]|$,

$$\begin{array}{rcl}[\,\vec{\mathcal{S}},\ (\mathbf{catch}\ v\ M')[\mathbf{val}\,\vec{R}/\vec{x},\vec{\overline{k}}/\vec{u}]\,] & = & [\,\vec{\mathcal{S}},\ \mathbf{catch}\ v\ M'[\mathbf{val}\,\vec{R}/\vec{x},\vec{\overline{k}}/\vec{u}]\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathbf{catch}\ v\ *,\ M'[\mathbf{val}\,\vec{R}/\vec{x},\vec{\overline{k}}/\vec{u},\overline{l}/v]\,] \\ & \overset{*}{\underset{l}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \mathbf{catch}\ v\ *,\ \vec{\mathcal{S}'},\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q\,]\end{array}$$

for some $\vec{\mathcal{S}'}$ and Q such that $\overline{Q} = V$ by the induction hypothesis.

Case 6: $M = \kappa\, v.\, M'$ *for some* v *and* M'. We can assume that $v \neq u_i$ for any i. First, suppose that $M[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} V$. By the definition of rewriting rules, for some V',

1. $M'[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} V'$ and $V = \kappa\, v.\, V'$, or
2. $M'[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ v\ V']$ and $V = \kappa\, v.\, \mathbf{throw}\ v\ V'$ for some $\mathcal{C}'$ which does not capture v.

In the first case, $len(M'[\vec{\overline{R}}/\vec{x}]) = len(M[\vec{\overline{R}}/\vec{x}])$ and $|M'[\vec{\overline{R}}/\vec{x}]| < |M[\vec{\overline{R}}/\vec{x}]|$. Therefore by the induction hypothesis, for some Q' such that $\overline{Q}' = V'$,

$$\begin{array}{l}[\,\vec{\mathcal{S}},\ (\kappa\, v.\, M')[\mathbf{val}\,\vec{R}/\vec{x},\vec{\overline{k}}/\vec{u}]\,] \\ \quad = [\,\vec{\mathcal{S}},\ \kappa\, v.\, M'[\mathbf{val}\,\vec{R}/\vec{x},\vec{\overline{k}}/\vec{u}]\,] \\ \quad \underset{l}{\Rightarrow} [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.\, *,\ M'[\mathbf{val}\,\vec{R}/\vec{x},\vec{\overline{k}}/\vec{u},\overline{l{+}1}/v]\,] \\ \quad \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.\, *,\ \mathbf{val}\ Q'\,] \\ \quad \underset{l}{\Rightarrow} [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \mathbf{val}\ (\kappa\, v.\,\mathbf{val}\ Q')\,] \\ \quad \underset{l}{\Rightarrow} [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\kappa\, v.\,\mathbf{val}\ Q')\,].\end{array}$$

Let Q be as $Q = \kappa\, v.\,\mathbf{val}\ Q'$, Then we get $\overline{Q} = \kappa\, v.\,\overline{Q}' = \kappa\, v.\, V' = V$. In the second case, $len(M'[\vec{\overline{R}}/\vec{x}]) \leq len(M[\vec{\overline{R}}/\vec{x}])$ and $|M'[\vec{\overline{R}}/\vec{x}]| < |M[\vec{\overline{R}}/\vec{x}]|$. Therefore,

$$\begin{array}{l}[\,\vec{\mathcal{S}},\ (\kappa\, v.\, M')[\mathbf{val}\,\vec{R}/\vec{x},\vec{\overline{k}}/\vec{u}]\,] \\ \quad \underset{l}{\Rightarrow} [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.\, *,\ M'[\mathbf{val}\,\vec{R}/\vec{x},\vec{\overline{k}}/\vec{u},\overline{l{+}1}/v]\,] \\ \quad \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.\, *,\ \vec{\mathcal{S}''},\ \mathbf{throw}\ \overline{l{+}1}\ *,\ \mathbf{val}\ Q'\,] \\ \quad \underset{l}{\Rightarrow} [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{val}\ Q'\,] \\ \quad \underset{l}{\Rightarrow} [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\kappa\, v.\,\mathbf{throw}\ v\ (\mathbf{val}\ Q'))\,]\end{array}$$

for some $\vec{\mathcal{S}''}$ and Q' such that $\overline{Q}' = V'$ by the induction hypothesis. Let Q be as $Q = \kappa\, v.\,\mathbf{throw}\ v$ $(\mathbf{val}\ Q')$, Then we get $\overline{Q} = \kappa\, v.\,\mathbf{throw}\ v\ \overline{Q}' = \kappa\, v.\,\mathbf{throw}\ v\ V' = V$.

Next suppose that $M[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}[\mathbf{throw}\ u_j\ V]$. By the definition of rewriting rules, there exists some $\mathcal{C}'$ such that $M'[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ V]$ and $\mathcal{C} = \kappa\, v.\,\mathcal{C}'$. Note that $\mathcal{C}'$ does not capture u_j. Since $len(M'[\vec{\overline{R}}/\vec{x}]) = len(M[\vec{\overline{R}}/\vec{x}])$ and $|M'[\vec{\overline{R}}/\vec{x}]| < |M[\vec{\overline{R}}/\vec{x}]|$,

$$\begin{array}{l}[\,\vec{\mathcal{S}},\ (\kappa\, v.\, M')[\mathbf{val}\,\vec{R}/\vec{x},\vec{\overline{k}}/\vec{u}]\,] \\ \quad \underset{l}{\Rightarrow} [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.\, *,\ M'[\mathbf{val}\,\vec{R}/\vec{x},\vec{\overline{k}}/\vec{u},\overline{l{+}1}/v]\,] \\ \quad \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.\, *,\ \vec{\mathcal{S}''},\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q\,]\end{array}$$

for some $\vec{\mathcal{S}''}$ and Q such that $\overline{Q} = V$ by the induction hypothesis.

Case 7: *$M = M'v$ for some M' and v.* First, suppose that $M[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} V$. By the definition of rewriting rules, for some w,

$$M'[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \kappa\, w.\, V.$$

Since $len(M'[\overline{\vec{R}}/\vec{x}]) < len(M[\overline{\vec{R}}/\vec{x}])$, by the induction hypothesis, for some Q such that $\overline{Q} = V$,

$$\begin{array}{rcl}
[\,\vec{\mathcal{S}},\ (M'v)[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] & = & [\,\vec{\mathcal{S}},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\; v[\overline{\vec{k}}/\vec{u}]\,] \\
& \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ *\,v[\overline{\vec{k}}/\vec{u}],\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \\
& \overset{*}{\underset{l}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ *\,v[\overline{\vec{k}}/\vec{u}],\ \mathbf{val}\ (\kappa\, w.\, \mathbf{val}\ Q)\,] \\
& \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ (\mathbf{val}\ Q)[v[\overline{\vec{k}}/\vec{u}]/w]\,] \\
& = & [\,\vec{\mathcal{S}},\ \mathbf{val}\ Q\,].
\end{array}$$

Next suppose that $M[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}[\mathbf{throw}\ u_j\ V]$. By the definition of rewriting rules,

1. $M'[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ V]$ and $\mathcal{C} = \mathcal{C}'v$ for some $\mathcal{C}'$, or
2. $M'[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \kappa\, w.\, \mathbf{throw}\ w\ V$ for some w and $v = u_j$.

In the first case, $len(M'[\overline{\vec{R}}/\vec{x}]) = len(M[\overline{\vec{R}}/\vec{x}])$ and $|M'[\overline{\vec{R}}/\vec{x}]| < |M[\overline{\vec{R}}/\vec{x}]|$. Therefore,

$$\begin{array}{l}
[\,\vec{\mathcal{S}},\ (M'v)[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \\
\quad \underset{l}{\Rightarrow}\ [\,\vec{\mathcal{S}},\ *\,v[\overline{\vec{k}}/\vec{u}],\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \\
\quad \overset{*}{\underset{l}{\Rightarrow}}\ [\,\vec{\mathcal{S}},\ *\,v[\overline{\vec{k}}/\vec{u}],\ \vec{\mathcal{S}}',\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q\,]
\end{array}$$

for some $\vec{\mathcal{S}}'$ and Q such that $\overline{Q} = V$ by the induction hypothesis. Similarly in the second case,

$$\begin{array}{rcl}
[\,\vec{\mathcal{S}},\ (M'v)[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] & = & [\,\vec{\mathcal{S}},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,\overline{k}_j\,] \\
& \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ *\,\overline{k}_j,\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \\
& \overset{*}{\underset{l}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ *\,\overline{k}_j,\ \mathbf{val}\ (\kappa\, w.\, \mathbf{throw}\ w\ (\mathbf{val}\ Q))\,] \\
& \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathbf{throw}\ \overline{k}_j\ (\mathbf{val}\ Q)\,] \\
& \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q\,].
\end{array}$$

Case 8: *$M = \lambda\, y.\, M'$ for some y and M'.* Since $M[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \lambda\, y.\, (M'[\overline{\vec{R}}/\vec{x}])$, we get $M[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\not\succ} \mathcal{C}[\mathbf{throw}\ u_j\ V]$. Suppose that $M[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} V$, i.e., $V = \lambda\, y.\, (M'[\overline{\vec{R}}/\vec{x}])$. Since $FTV(\lambda\, y.\, M') = \{\}$, we get $(\lambda\, y.\, M')[\mathbf{val}\ \vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}] = \lambda\, y.\, (M'[\mathbf{val}\ \vec{R}/\vec{x}])$. Therefore, $[\,\vec{\mathcal{S}},\ M[\mathbf{val}\ \vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \underset{l}{\Rightarrow} [\,\vec{\mathcal{S}},\ \mathbf{val}\ Q\,]$, where $Q = \lambda\, y.\, (M'[\mathbf{val}\,\vec{R}/\vec{x}])$, and $\overline{Q} = \lambda\, y.\, (M'[\overline{\vec{R}}/\vec{x}])$ by Proposition 3.1.8.

Case 9: *M has one of other forms.* Similar. □

Lemma 3.2.5 *Let $\vec{\mathcal{S}}$ be a sequence of context segments whose length is l, and let e be an internal term. If $[\,\vec{\mathcal{S}},\ e\,] \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}}',\ e'\,]$ for some $\vec{\mathcal{S}}'$ and e', then for any tag constant $\overline{m}$ occurring in $\vec{\mathcal{S}}'$ or e', $m < l$ implies that $\overline{m}$ occurs in e.*

Proof. By induction on the length of the path of $[\,\vec{\mathcal{S}},\ e\,] \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}}',\ e'\,]$. Each step is obvious from the definition of transition rules. □

Lemma 3.2.6 *Let $\vec{\mathcal{S}}$ be a sequence of context segments whose length is l. Let $u_1, \ldots, u_n$ be tag variables, and $\overline{k}_1, \ldots, \overline{k}_n$ tag constants. Let M be a term such that $FIV(M) = \{\}$ and $FTV(M) \subset \{u_1, \ldots, u_n\}$. If*

$$[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\Rightarrow} [\ \vec{\mathcal{S}'},\ e\ ]$$

for some $\mathcal{S}'$ and e, then

1. $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \underset{l}{\stackrel{*}{\Rightarrow}} [\ \vec{\mathcal{S}'},\ e\ ]$,
2. $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \underset{l}{\stackrel{*}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ]$ *for some Q, or*
3. $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \underset{l}{\stackrel{*}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q\ ]$ *and $k_j < l$ for some Q and j.*

Proof. Assume that $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \not\underset{l}{\stackrel{*}{\Rightarrow}} [\ \vec{\mathcal{S}'},\ e\ ]$, and let $[\ \vec{\mathcal{S}''},\ e'\ ]$ be the final state of $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ]$ w.r.t. $\underset{l}{\Rightarrow}$, which is not a final state w.r.t. $\Rightarrow$ since $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\Rightarrow} [\ \vec{\mathcal{S}'},\ e\ ]$. Therefore, a transition rule is applicable to the state. By the definition of transition rules,

1. $[\ \vec{\mathcal{S}''},\ e'\ ] = [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ]$ for some Q, or
2. $[\ \vec{\mathcal{S}''},\ e'\ ] = [\ \vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{m}\ *,\ \mathbf{val}\ Q\ ]$ for some Q and m such that $m < l$.

We get $m = k_j$ for some j by Lemma 3.2.5 in the latter case. □

Lemma 3.2.7 *Let $\vec{\mathcal{S}}$ and $\vec{\mathcal{S}'}$ be sequences of context segments whose lengths are l and m, respectively. Let $u_1, \ldots, u_n$ be tag variables, and $\overline{k}_1, \ldots, \overline{k}_n$ tag constants. Let M be a term such that $FIV(M) = \{\}$ and $FTV(M) \subset \{u_1, \ldots, u_n\}$. If*

$$[\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\Rightarrow} [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ]$$

for some Q, then

1. $[\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \underset{l+m}{\stackrel{*}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \mathbf{val}\ Q'\ ]$ *for some Q', or*
2. $[\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \underset{l+m}{\stackrel{*}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \ldots,\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q'\ ]$ *and $l \leq k_j < l + m$ for some Q' and j.*

Proof. Suppose that $[\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\Rightarrow} [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ]$. By Lemma 3.2.6,

1. $[\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \underset{l+m}{\stackrel{*}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ]$,
2. $[\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \underset{l+m}{\stackrel{*}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \mathbf{val}\ Q'\ ]$ for some Q', or
3. $[\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \underset{l+m}{\stackrel{*}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \ldots,\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q'\ ]$ and $k_j < l + m$ for some Q' and j.

In the first case, we get $m = 0$ by the definition of $\underset{l+m}{\stackrel{*}{\Rightarrow}}$. Therefore, $[\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ] = [\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \mathbf{val}\ Q\ ]$. Trivial in the second case. In the third case, it is enough to show that $l \leq k_j$. Since $[\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ]$ is a final state w.r.t. $\underset{l}{\Rightarrow}$, we get that $[\ \vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \ldots,\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q'\ ]$ is not a final state w.r.t. $\underset{l}{\Rightarrow}$. Therefore we get $l \leq k_j$. □

Lemma 3.2.8 *Let $\vec{\mathcal{S}}$ be a sequence of context segments whose length is l. Let $u_1, \ldots, u_n$ be tag variables, and $\overline{k}_1, \ldots, \overline{k}_n$ tag constants. Let M be a term such that $FIV(M) = \{\}$ and $FTV(M) \subset \{u_1, \ldots, u_n\}$. If $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\Rightarrow} [\ \vec{\mathcal{S}'},\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\ ]$ for some $\vec{\mathcal{S}'}$, Q and q such that $q < l$, then*

1. *$[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q'\ ]$ for some Q', or*
2. *$[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\ ]$.*

Proof. Suppose that $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\Rightarrow} [\ \vec{\mathcal{S}'},\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\ ]$ and $q < l$. By Lemma 3.2.6,

1. $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}'},\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\ ]$,
2. $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q'\ ]$ for some Q', or
3. $[\ \vec{\mathcal{S}},\ M[\overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]\ ] \stackrel{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q'\ ]$ and $k_j < l$ for some Q' and j.

In the first case, let m be the length of $\vec{\mathcal{S}'}$. Trivial if $l \leq m$. Otherwise we get $[\ \vec{\mathcal{S}'},\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\ ] = [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ]$ from the definition of $\underset{l}{\Rightarrow}$. Trivial in the the second case. In the third case, since $k_j < l$, $[\ \vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q'\ ]$ is a final state w.r.t. $\underset{l}{\Rightarrow}$ as well as $[\ \vec{\mathcal{S}}, \ldots,\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\ ]$. Therefore, $k_j = q$ and $Q' = Q$. □

Theorem 3.2.9 *Let $\vec{\mathcal{S}}$ be a sequence of context segments whose lengths is l. Let $x_1, \ldots, x_m$ be individual variables, and $R_1, \ldots, R_m$ internal values. Let $u_1, \ldots, u_n$ be tag variables, and $\overline{k}_1, \ldots, \overline{k}_n$ tag constants such that $k_i \leq l$ for any i. Let M be a term such that $FIV(M) \subset \{x_1, \ldots, x_m\}$ and $FTV(M) \subset \{u_1, \ldots, u_n\}$. Let $M[\overline{\vec{R}}/\vec{x}]$ and $M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]$ be abbreviations for $M[\overline{R}_1/x_1, \ldots, \overline{R}_m/x_m]$ and $M[\mathbf{val}\ R_1/x_1, \ldots, \mathbf{val}\ R_m/x_m, \overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]$, respectively. Let Q be an internal value.*

1. *If $[\ \vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\ ] \stackrel{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ]$, then*
$$M[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} \overline{Q} \quad or \quad M[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}]$$
for some j and $\mathcal{C}$ such that $k_j = l$ and $\mathcal{C}$ does not capture u_j.
2. *If $[\ \vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\ ] \stackrel{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\ ]$ for some q such that $q < l$, then*
$$M[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}]$$
for some j and $\mathcal{C}$ such that $q = k_j$ and $\mathcal{C}$ does not capture u_j.

Proof. By induction on the lengths of the following transition paths.

$$[\ \vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\ ] \ \stackrel{*}{\underset{l}{\Rightarrow}}\ [\ \vec{\mathcal{S}},\ \mathbf{val}\ Q\ ] \tag{3.1}$$
$$[\ \vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\ ] \ \stackrel{*}{\underset{l}{\Rightarrow}}\ [\ \vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\ ] \tag{3.2}$$

By cases according to the form of M.

Case 1: *M is an individual constant.* Trivial since $[\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] = [\,\vec{\mathcal{S}},\ M\,] \underset{l}{\Rightarrow} [\,\vec{\mathcal{S}},\ \mathbf{val}\,M\,]$ and $M \underset{\mathrm{CBV}}{\succ} M = \overline{M}$.

Case 2: *M is an individual variable.* Suppose that

$$[\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \ldots,\ \mathbf{val}\,Q\,].$$

Since $M = x_i$ for some i, we get $M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}] = \mathbf{val}\ R_i$. That is, $[\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,]$ is a final state w.r.t. $\underset{l}{\Rightarrow}$. Therefore (3.2) does not hold. Suppose that (3.1) holds. In this case, $Q = R_i$. Therefore, $M[\overline{\vec{R}}/\vec{x}] = \overline{R}_i \underset{\mathrm{CBV}}{\succ} \overline{R}_i = \overline{Q}$.

Case 3: *$M = \mathbf{let}\ y{=}M_1.\ M_2$ for some y, M_1 and M_2.* We can assume that $y \not\equiv x_i$ for any i. Let $\mathcal{S}_{l+1}$ be as $\mathcal{S}_{l+1} = \mathbf{let}\ y{=}{*}.\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]$. By the definition of transition rules,

$$\begin{array}{rl} & [\,\vec{\mathcal{S}},\ (\mathbf{let}\ y{=}M_1.\ M_2)[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \\ = & [\,\vec{\mathcal{S}},\ \mathbf{let}\ y{=}M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}].\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \\ \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,]. \end{array}$$

First, suppose (3.1), i.e., $[\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathbf{val}\,Q\,]$. By Lemma 3.2.7,

1. $[\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \mathbf{val}\,Q'\,]$ for some Q', or
2. $[\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \ldots,\ \mathbf{throw}\ \overline{k}_i\ {*},\ \mathbf{val}\,Q'\,]$ and $k_i = l$ for some Q' and i.

In the first case, since $k_j < l+1$ for any j, we get $M_1[\overline{\vec{R}}/\vec{x}] \underset{\mathrm{CBV}}{\succ} \overline{Q}'$ by the induction hypothesis. On the other hand,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \\ & \overset{*}{\underset{l+1}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \mathbf{val}\,Q'\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \mathbf{val}\,Q'/y, \overline{\vec{k}}/\vec{u}]\,]. \end{array}$$

Therefore, we get $[\,\vec{\mathcal{S}},\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \mathbf{val}\,Q'/y, \overline{\vec{k}}/\vec{u}]\,] \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathbf{val}\,Q\,]$ by (3.1). By the induction hypothesis again, we get $M_2[\overline{\vec{R}}/\vec{x}, \overline{Q}'/y] \underset{\mathrm{CBV}}{\succ} \overline{Q}$ or $M_2[\overline{\vec{R}}/\vec{x}, \overline{Q}'/y] \underset{\mathrm{CBV}}{\succ} \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}]$ for some j and $\mathcal{C}$ such that $k_j = l$ and $\mathcal{C}$ does not capture u_j. That is,

$$\begin{array}{rcl} M[\overline{\vec{R}}/\vec{x}] & = & \mathbf{let}\ y{=}M_1[\overline{\vec{R}}/\vec{x}].\ M_2[\overline{\vec{R}}/\vec{x}] \\ & \overset{*}{\underset{\mathrm{CBV}}{\longrightarrow}} & \mathbf{let}\ y{=}\overline{Q}'.\ M_2[\overline{\vec{R}}/\vec{x}] \\ & \underset{\mathrm{CBV}}{\longrightarrow} & M_2[\overline{\vec{R}}/\vec{x}, \overline{Q}'/y] \\ & \underset{\mathrm{CBV}}{\succ} & \overline{Q}\ \ \text{or}\ \ \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}]. \end{array}$$

In the second case,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \\ & \overset{*}{\underset{l+1}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \ldots,\ \mathbf{throw}\ \overline{l}\ {*},\ \mathbf{val}\,Q'\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathbf{val}\,Q'\,]. \end{array}$$

Since $[\ \vec{\mathcal{S}},\ \mathbf{val}\ Q'\ ]$ is a final state w.r.t $\underset{l}{\Rightarrow}$, we get $Q' = Q$ from (3.1). That is,

$$[\ \vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\ ] \overset{*}{\underset{l+1}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \ldots,\ \mathbf{throw}\ \overline{l} *,\ \mathbf{val}\ Q\ ].$$

By the induction hypothesis, we get $M_1[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}]$ for some j and $\mathcal{C}'$ such that $l = k_j$ and $\mathcal{C}'$ does not capture u_j. Therefore,

$$\begin{array}{rcl} M[\overline{\vec{R}}/\vec{x}] & = & \mathbf{let}\ y{=}M_1[\overline{\vec{R}}/\vec{x}].\ M_2[\overline{\vec{R}}/\vec{x}] \\ & \overset{*}{\underset{\text{CBV}}{\longrightarrow}} & \mathbf{let}\ y{=}\mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}].\ M_2[\overline{\vec{R}}/\vec{x}] \\ & = & \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}], \end{array}$$

where $\mathcal{C} = \mathbf{let}\ y{=}\mathcal{C}'.\ M_2[\overline{\vec{R}}/\vec{x}]$, which does not capture u_j.

Next suppose that $q < l$ and (3.2), i.e.,

$$\begin{array}{rcl} [\ \vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\ ] & \underset{l}{\Rightarrow} & [\ \vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\ ] \\ & \overset{*}{\underset{l}{\Rightarrow}} & [\ \vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{q} *,\ \mathbf{val}\ Q\ ]. \end{array}$$

By Lemma 3.2.8,

1. $[\ \vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\ ] \overset{*}{\underset{l+1}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \mathbf{val}\ Q'\ ]$ for some Q', or

2. $[\ \vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M_1[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\ ] \overset{*}{\underset{l+1}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \ldots,\ \mathbf{throw}\ \overline{q} *,\ \mathbf{val}\ Q\ ]$.

In the first case, since $k_j < l+1$ for any j, we get $M_1[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \overline{Q}'$ by the induction hypothesis. On the other hand,

$$\begin{array}{rcl} [\ \vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\ ] & \overset{*}{\underset{l}{\Rightarrow}} & [\ \vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \mathbf{val}\ Q'\ ] \\ & \underset{l}{\Rightarrow} & [\ \vec{\mathcal{S}},\ M_2[\mathbf{val}\,\vec{R}/\vec{x}, \mathbf{val}\ Q'/y, \overline{\vec{k}}/\vec{u}]\ ]. \end{array}$$

Therefore, $[\ \vec{\mathcal{S}},\ M_2[\mathbf{val}\ \vec{R}/\vec{x}, \mathbf{val}\ Q'/y, \overline{\vec{k}}/\vec{u}]\ ] \overset{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{q} *,\ \mathbf{val}\ Q\ ]$ by (3.2). By the induction hypothesis, we get $M_2[\overline{\vec{R}}/\vec{x}, \overline{Q}/y] \underset{\text{CBV}}{\succ} \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}]$ for some j and $\mathcal{C}$ such that $q = k_j$ and $\mathcal{C}$ does not capture u_j. Therefore,

$$\begin{array}{rcl} M[\overline{\vec{R}}/\vec{x}] & = & \mathbf{let}\ y{=}M_1[\overline{\vec{R}}/\vec{x}].\ M_2[\overline{\vec{R}}/\vec{x}] \\ & \overset{*}{\underset{\text{CBV}}{\longrightarrow}} & \mathbf{let}\ y{=}\overline{Q}'.\ M_2[\overline{\vec{R}}/\vec{x}] \\ & \underset{\text{CBV}}{\longrightarrow} & M_2[\overline{\vec{R}}/\vec{x}, \overline{Q}'/y] \\ & \underset{\text{CBV}}{\succ} & \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}]. \end{array}$$

In the second case, we get $M_1[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}]$ for some j and $\mathcal{C}'$ such that $q = k_j$ and $\mathcal{C}'$ does not capture u_j, by the induction hypothesis. Therefore,

$$\begin{array}{rcl} M[\overline{\vec{R}}/\vec{x}] & = & \mathbf{let}\ y{=}M_1[\overline{\vec{R}}/\vec{x}].\ M_2[\overline{\vec{R}}/\vec{x}] \\ & \underset{\text{CBV}}{\succ} & \mathbf{let}\ y{=}\mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}].\ M_2[\overline{\vec{R}}/\vec{x}] \\ & = & \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}], \end{array}$$

where $\mathcal{C} = \mathbf{let}\ y{=}\mathcal{C}'.\ M_2[\overline{\vec{R}}/\vec{x}]$, which does not capture u_j.

Case 4: $M = \mathbf{throw}\ v\ M'$ *for some* v *and* M'. Since $FTV(M) \subset \{u_1, \ldots, u_n\}$, we get $v = u_p$ for some p. Let $\mathcal{S}_{l+1}$ be as $\mathcal{S}_{l+1} = \mathbf{throw}\ \overline{k}_p\ *$. By the definition of transition rules,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ (\mathbf{throw}\ v\ M')[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] & = & [\,\vec{\mathcal{S}},\ \mathbf{throw}\ \overline{k}_p\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,]. \end{array}$$

First, suppose (3.1), i.e., $[\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathbf{val}\ Q\,]$. By Lemma 3.2.7,

1. $[\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \mathbf{val}\ Q'\,]$ for some Q', or

2. $[\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \ldots,\ \mathbf{throw}\ \overline{k}_i\ *,\ \mathbf{val}\ Q'\,]$ and $k_i = l$ for some Q' and i.

In the first case, since $k_j < l+1$ for any j, we get $M'[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \overline{Q}'$ by the induction hypothesis. Therefore,

$$M[\vec{\overline{R}}/\vec{x}] \quad = \quad \mathbf{throw}\ u_p\ M'[\vec{\overline{R}}/\vec{x}] \quad \underset{\text{CBV}}{\succ} \quad \mathbf{throw}\ u_p\ \overline{Q}'.$$

On the other hand, $[\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \mathbf{val}\ Q'\,] \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathbf{val}\ Q\,]$ by (3.1). Therefore, we get $k_p = l$ and $Q = Q'$. In the second case, since $k_i = l$,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \\ & \overset{*}{\underset{l+1}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \ldots,\ \mathbf{throw}\ \overline{l}\ *,\ \mathbf{val}\ Q'\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathbf{val}\ Q'\,]. \end{array}$$

Therefore, $Q' = Q$ by (3.1), i.e.,

$$[\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \ldots,\ \mathbf{throw}\ \overline{l}\ *,\ \mathbf{val}\ Q\,].$$

By the induction hypothesis, we get $M'[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}]$ for some j and $\mathcal{C}'$ such that $l = k_j$ and $\mathcal{C}'$ does not capture u_j. Therefore,

$$\begin{array}{rcl} M[\vec{\overline{R}}/\vec{x}] & = & \mathbf{throw}\ u_p\ M'[\vec{\overline{R}}/\vec{x}] \\ & \xrightarrow[\text{CBV}]{*} & \mathbf{throw}\ u_p\ \mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}] \\ & = & \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}], \end{array}$$

where $\mathcal{C} = \mathbf{throw}\ u_p\ \mathcal{C}'$, which does not capture u_j.

Next suppose that $q < l$ and (3.2), i.e.,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \\ & \overset{*}{\underset{l}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\,]. \end{array}$$

By Lemma 3.2.8,

1. $[\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \mathbf{val}\ Q'\,]$ for some Q', or

2. $[\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \ldots,\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\,]$.

In the first case, since $k_j < l+1$ for any j, we get $M'[\vec{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \overline{Q}'$ by the induction hypothesis. Therefore,

$$M[\vec{\vec{R}}/\vec{x}] \;=\; \textbf{throw}\ u_p\ M'[\vec{\vec{R}}/\vec{x}] \;\underset{\text{CBV}}{\succ}\; \textbf{throw}\ u_p\ \overline{Q}'.$$

On the other hand, $[\ \vec{\mathcal{S}},\ \mathcal{S}_{l+1},\ \textbf{val}\ Q'\ ] \overset{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \ldots,\ \textbf{throw}\ \overline{q}\ *,\ \textbf{val}\ Q\ ]$ by (3.2). Therefore we get $q = k_p$ and $Q' = Q$. In the second case, we get $M'[\vec{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\textbf{throw}\ u_j\ \overline{Q}]$ for some j and $\mathcal{C}'$ such that $q = k_j$ and $\mathcal{C}'$ does not capture u_j, by the induction hypothesis. Therefore,

$$\begin{array}{rcl} M[\vec{\vec{R}}/\vec{x}] & = & \textbf{throw}\ u_p\ M'[\vec{\vec{R}}/\vec{x}] \\ & \underset{\text{CBV}}{\succ} & \textbf{throw}\ u_p\ \mathcal{C}'[\textbf{throw}\ u_j\ \overline{Q}] \\ & = & \mathcal{C}[\textbf{throw}\ u_j\ \overline{Q}], \end{array}$$

where $\mathcal{C} = \textbf{throw}\ u_p\ \mathcal{C}'$, which does not capture u_j.

Case 5: $M = \textbf{catch}\ v\ M'$ *for some* v *and* M'. We can assume that $v \neq u_i$ for any i. By the definition of transition rules,

$$\begin{array}{rcl} [\ \vec{\mathcal{S}},\ (\textbf{catch}\ v\ M')[\textbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ] & = & [\ \vec{\mathcal{S}},\ \textbf{catch}\ v\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ] \\ & \underset{l}{\Rightarrow} & [\ \vec{\mathcal{S}},\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}, \overline{l}/v]\ ]. \end{array}$$

First, suppose (3.1), i.e., $[\ \vec{\mathcal{S}},\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}, \overline{l}/v]\ ] \overset{*}{\underset{l}{\Rightarrow}} [\ \vec{\mathcal{S}},\ \textbf{val}\ Q\ ]$. By the induction hypothesis,

1. $M'[\vec{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \overline{Q}$,
2. $M'[\vec{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\textbf{throw}\ u_j\ \overline{Q}]$ and $k_j = l$ for some j and $\mathcal{C}'$ which does not capture u_j, or
3. $M'[\vec{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\textbf{throw}\ v\ \overline{Q}]$ for some $\mathcal{C}'$ which does not capture v.

In the first case,

$$M[\vec{\vec{R}}/\vec{x}] \;\xrightarrow[\text{CBV}]{*}\; \textbf{catch}\ v\ \overline{Q} \;\underset{\text{CBV}}{\succ}\; \overline{Q}.$$

In the second case,

$$M[\vec{\vec{R}}/\vec{x}] \;\underset{\text{CBV}}{\succ}\; \textbf{catch}\ v\ \mathcal{C}'[\textbf{throw}\ u_j\ \overline{Q}] \;=\; \mathcal{C}[\textbf{throw}\ u_j\ \overline{Q}],$$

where $\mathcal{C} = \textbf{catch}\ v\ \mathcal{C}'$, which does not capture u_j. In the third case,

$$M[\vec{\vec{R}}/\vec{x}] \;\xrightarrow[\text{CBV}]{*}\; \textbf{catch}\ v\ \mathcal{C}'[\textbf{throw}\ v\ \overline{Q}] \;\xrightarrow[\text{CBV}]{*}\; \textbf{catch}\ v\ (\textbf{throw}\ v\ \overline{Q}) \;\xrightarrow[\text{CBV}]{}\; \overline{Q}.$$

Next suppose that $q < l$ and (3.2), i.e.,

$$\begin{array}{rcl} [\ \vec{\mathcal{S}},\ M[\textbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ] & \underset{l}{\Rightarrow} & [\ \vec{\mathcal{S}},\ M'[\textbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}, \overline{l}/v]\ ] \\ & \overset{*}{\underset{l}{\Rightarrow}} & [\ \vec{\mathcal{S}},\ \ldots,\ \textbf{throw}\ \overline{q}\ *,\ \textbf{val}\ Q\ ]. \end{array}$$

By the induction hypothesis, $M'[\vec{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\textbf{catch}\ u_j\ \overline{Q}]$ and $k_j = q$ for some j and $\mathcal{C}'$ which does not capture u_j. Therefore,

$$M[\vec{\vec{R}}/\vec{x}] \;\underset{\text{CBV}}{\succ}\; \textbf{catch}\ v\ \mathcal{C}'[\textbf{throw}\ u_j\ \overline{Q}] \;=\; \mathcal{C}[\textbf{throw}\ u_j\ \overline{Q}],$$

where $\mathcal{C} = \textbf{catch}\ v\ \mathcal{C}'$, which does not capture u_j.

Case 6: *$M = \kappa v. M'$ for some v and M'.* We can assume that $v \neq u_i$ for any i. By the definition of transition rules,

$$\begin{array}{rl} [\,\vec{\mathcal{S}},\ (\kappa\, v.\, M')[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] & \\ = & [\,\vec{\mathcal{S}},\ \kappa\, v.\, M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \\ \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.\,*,\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}, \overline{l+1}/v]\,]. \end{array}$$

Let $\vec{\mathcal{S}'}$ be as $\vec{\mathcal{S}'} = \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.\,*$. First, suppose (3.1), i.e., $[\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}, \overline{l+1}/v]\,] \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathbf{val}\ Q\,]$. By Lemma 3.2.7,

1. $[\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}, \overline{l+1}/v]\,] \overset{*}{\underset{l+3}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \mathbf{val}\ Q'\,]$ for some Q',
2. $[\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}, \overline{l+1}/v]\,] \overset{*}{\underset{l+3}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \ldots,\ \mathbf{throw}\ \overline{k}_i\ *,\ \mathbf{val}\ Q'\,]$ and $k_i = l$ for some Q' and i, or
3. $[\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}, \overline{l+1}/v]\,] \overset{*}{\underset{l+3}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \ldots,\ \mathbf{throw}\ \overline{l+1}\ *,\ \mathbf{val}\ Q'\,]$ and for some Q'.

In the first case, since $l+1$, $k_j < l+3$ for any j, we get $M'[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \overline{Q}'$ by the induction hypothesis. Therefore,

$$M[\vec{\overline{R}}/\vec{x}] \quad = \quad \kappa\, v.\, M'[\vec{\overline{R}}/\vec{x}] \quad \underset{\text{CBV}}{\succ} \quad \kappa\, v.\,\overline{Q}'.$$

On the other hand,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \mathbf{val}\ Q'\,] & = & [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.\,*,\ \mathbf{val}\ Q'\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \mathbf{val}\ (\kappa\, v.\,\mathbf{val}\ Q')\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\kappa\, v.\,\mathbf{val}\ Q')\,]. \end{array}$$

Therefore, we get $Q = \kappa\, v.\,\mathbf{val}\ Q'$ by (3.1), i.e., $\overline{Q} = \kappa\, v.\,\overline{Q}'$. In the second case,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}, \overline{l+1}/v]\,] \\ & \overset{*}{\underset{l+3}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \ldots,\ \mathbf{throw}\ \overline{l}\ *,\ \mathbf{val}\ Q'\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathbf{val}\ Q'\,]. \end{array}$$

Therefore $Q' = Q$ by (3.1), and

$$[\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}, \overline{l+1}/v]\,] \quad \overset{*}{\underset{l+3}{\Rightarrow}} \quad [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \ldots,\ \mathbf{throw}\ \overline{l}\ *,\ \mathbf{val}\ Q\,].$$

By the induction hypothesis, we get $M'[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}]$ for some j and $\mathcal{C}'$ such that $l = k_j$ and $\mathcal{C}'$ does not capture u_j. That is,

$$\begin{array}{rcl} M[\vec{\overline{R}}/\vec{x}] & = & \kappa\, v.\, M'[\vec{\overline{R}}/\vec{x}] \\ & \overset{*}{\underset{\text{CBV}}{\longrightarrow}} & \kappa\, v.\,\mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}] \\ & = & \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}], \end{array}$$

where $\mathcal{C} = \kappa\, v.\,\mathcal{C}'$, which does not capture u_j. In the third case, by the induction hypothesis, we get $M'[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ v\ \overline{Q}']$ for some $\mathcal{C}'$ which does not capture v. Therefore,

$$\begin{array}{rcl} M[\vec{\overline{R}}/\vec{x}] & = & \kappa\, v.\, M'[\vec{\overline{R}}/\vec{x}] \\ & \overset{*}{\underset{\text{CBV}}{\longrightarrow}} & \kappa\, v.\,\mathcal{C}'[\mathbf{throw}\ v\ \overline{Q}'] \\ & \underset{\text{CBV}}{\succ} & \kappa\, v.\,\mathbf{throw}\ v\ \overline{Q}'. \end{array}$$

On the other hand,

$$\begin{array}{ll} [\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u}]\,] & \\ \quad \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u},\overline{l{+}1}/v]\,] \\ \quad \overset{*}{\underset{l+3}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.*,\ \ldots,\ \mathbf{throw}\ \overline{l{+}1}\ *,\ \mathbf{val}\ Q'\,] \\ \quad \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{val}\ Q'\,] \\ \quad \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\kappa\, v.\,\mathbf{throw}\ v\ \mathbf{val}\ Q')\,]. \end{array}$$

Therefore we get $Q = \kappa\, v.\,\mathbf{throw}\ v\ (\mathbf{val}\ Q')$ by (3.1), i.e., $\overline{Q} = \kappa\, v.\,\mathbf{throw}\ v\ \overline{Q}'$.

Next, suppose that $q < l$ and (3.2), i.e.,

$$\begin{array}{lll} [\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u}]\,] & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u},\overline{l{+}1}/v]\,] \\ & \overset{*}{\underset{l}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\,]. \end{array}$$

By Lemma 3.2.8,

1. $[\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u},\overline{l{+}1}/v]\,] \overset{*}{\underset{l+3}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \mathbf{val}\ Q'\,]$ for some Q', or

2. $[\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u},\overline{l{+}1}/v]\,] \overset{*}{\underset{l+3}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ \ldots,\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\,]$.

We need not consider the first case, because in this case,

$$\begin{array}{ll} [\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u}]\,] & \\ \quad \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \vec{\mathcal{S}'},\ M'[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u},\overline{l{+}1}/v]\,] \\ \quad \overset{*}{\underset{l+3}{\Rightarrow}} & [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{throw}\ \overline{l}\ *,\ \kappa\, v.*,\ \ldots,\ \mathbf{throw}\ \overline{l{+}1}\ *,\ \mathbf{val}\ Q'\,] \\ \quad \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \kappa\, v.\,\mathbf{throw}\ v\ *,\ \mathbf{val}\ Q'\,] \\ \quad \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\kappa\, v.\,\mathbf{throw}\ v\ (\mathbf{val}\ Q'))\,], \end{array}$$

and this contradicts (3.2). In the second case, we get $M'[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}]$ for some j and $\mathcal{C}$ such that $q = k_j$ and $\mathcal{C}'$ does not capture u_j by the induction hypothesis. Therefore,

$$M[\overline{\vec{R}}/\vec{x}] \quad \underset{\text{CBV}}{\succ} \quad \kappa\, v.\,\mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}] \quad = \quad \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}],$$

where $\mathcal{C} = \kappa\, v.\,\mathcal{C}'$, which does not capture u_j.

Case 7: *$M = M'v$ for some M' and v.* Since $FTV(M) \subset \{u_1, \ldots, u_n\}$, we get $v = u_p$ for some p. By the definition of transition rules,

$$\begin{array}{lll} [\,\vec{\mathcal{S}},\ (M'v)[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u}]\,] & = & [\,\vec{\mathcal{S}},\ M'[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u}]\,\overline{k}_p\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\ *\,\overline{k}_p,\ M'[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u}]\,]. \end{array}$$

First, suppose (3.1), i.e., $[\,\vec{\mathcal{S}},\ *\,\overline{k}_p,\ M'[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u}]\,] \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \mathbf{val}\ Q\,]$. By Lemma 3.2.7,

1. $[\,\vec{\mathcal{S}},\ *\,\overline{k}_p,\ M'[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ *\,\overline{k}_p,\ \mathbf{val}\ Q'\,]$ for some Q', or

2. $[\,\vec{\mathcal{S}},\ *\,\overline{k}_p,\ M'[\mathbf{val}\,\vec{R}/\vec{x},\overline{\vec{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ *\,\overline{k}_p,\ \ldots,\ \mathbf{throw}\ \overline{k}_i\ *,\ \mathbf{val}\ Q'\,]$ and $k_i = l$ for some Q' and i.

In the first case, since $k_j < l+1$ for any j, we get $M'[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \overline{Q}'$ by the induction hypothesis. Therefore,

$$M[\vec{\overline{R}}/\vec{x}] \;=\; M'[\vec{\overline{R}}/\vec{x}]\,u_p \;\xrightarrow[\text{CBV}]{*}\; \overline{Q}'\,u_p.$$

On the other hand, we get $Q' = \kappa\, w.\, e$ for some w and e from (3.1). Moreover, since Q' is an internal value, for some Q'',

1a. $Q' = \kappa\, w.\, \mathbf{val}\, Q''$, or

1b. $Q' = \kappa\, w.\, \mathbf{throw}\; w\; (\mathbf{val}\, Q'')$.

In the case of 1a,

$$[\,\vec{\mathcal{S}},\; *\overline{k}_p,\; \mathbf{val}\, Q'\,] \;=\; [\,\vec{\mathcal{S}},\; *\overline{k}_p,\; \mathbf{val}\,(\kappa\, w.\, \mathbf{val}\, Q'')\,] \;\underset{l}{\Rightarrow}\; [\,\vec{\mathcal{S}},\; \mathbf{val}\, Q''\,].$$

We so get $Q = Q''$ from (3.1). Therefore $\overline{Q}' = \kappa\, w.\, \overline{Q}$, and

$$M[\vec{\overline{R}}/\vec{x}] \;\xrightarrow[\text{CBV}]{*}\; \overline{Q}'\,u_p \;=\; (\kappa\, w.\, \overline{Q})\,u_p \;\underset{\text{CBV}}{\succ}\; \overline{Q}.$$

In the case of 1b,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\; *\overline{k}_p,\; \mathbf{val}\, Q'\,] & = & [\,\vec{\mathcal{S}},\; *\overline{k}_p,\; \mathbf{val}\,(\kappa\, w.\, \mathbf{throw}\; w\; (\mathbf{val}\, Q''))\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\; \mathbf{throw}\; \overline{k}_p\; (\mathbf{val}\, Q'')\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\; \mathbf{throw}\; \overline{k}_p\; *,\; \mathbf{val}\, Q''\,]. \end{array}$$

We so get $k_p = l$ and $Q'' = Q$ from (3.1). Therefore $\overline{Q}' = \kappa\, w.\, \mathbf{throw}\; w\; \overline{Q}$, and

$$M[\vec{\overline{R}}/\vec{x}] \;\xrightarrow[\text{CBV}]{*}\; \overline{Q}'\,u_p. \;=\; (\kappa\, w.\, \mathbf{throw}\; w\; \overline{Q})\,u_p \;\underset{\text{CBV}}{\succ}\; \mathbf{throw}\; u_p\; \overline{Q}.$$

In the second case,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\; M[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\; *\overline{k}_p,\; M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \\ & \overset{*}{\underset{l+1}{\Rightarrow}} & [\,\vec{\mathcal{S}},\; *\overline{k}_p,\; \ldots,\; \mathbf{throw}\; \overline{l}\; *,\; \mathbf{val}\, Q'\,] \\ & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\; \mathbf{val}\, Q'\,]. \end{array}$$

We so get $Q' = Q$ from (3.1). Therefore,

$$[\,\vec{\mathcal{S}},\; *\overline{k}_p,\; M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \;\overset{*}{\underset{l+1}{\Rightarrow}}\; [\,\vec{\mathcal{S}},\; *\overline{k}_p,\; \ldots,\; \mathbf{throw}\; \overline{l}\; *,\; \mathbf{val}\, Q\,].$$

By the induction hypothesis, we get $M'[\vec{\overline{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\; u_j\; \overline{Q}]$ for some j and $\mathcal{C}'$ such that $l = k_j$ and $\mathcal{C}'$ does not capture u_j. That is,

$$\begin{array}{rcl} M[\vec{\overline{R}}/\vec{x}] & = & M'[\vec{\overline{R}}/\vec{x}]\,\overline{k}_p \\ & \xrightarrow[\text{CBV}]{*} & \mathcal{C}'[\mathbf{throw}\; u_j\; \overline{Q}]\,\overline{k}_p \\ & = & \mathcal{C}[\mathbf{throw}\; u_j\; \overline{Q}], \end{array}$$

where $\mathcal{C} = \mathcal{C}'\,\overline{k}_p$, which does not capture u_j.

Next, suppose that $q < l$ and (3.2), i.e.,

$$\begin{array}{rcl} [\,\vec{\mathcal{S}},\; M[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] & \underset{l}{\Rightarrow} & [\,\vec{\mathcal{S}},\; *\overline{k}_p,\; M'[\mathbf{val}\,\vec{R}/\vec{x}, \vec{\overline{k}}/\vec{u}]\,] \\ & \overset{*}{\underset{l}{\Rightarrow}} & [\,\vec{\mathcal{S}},\; \ldots,\; \mathbf{throw}\; \overline{q}\; *,\; \mathbf{val}\, Q\,]. \end{array}$$

By Lemma 3.2.8,

1. $[\,\vec{\mathcal{S}},\ *\overline{k}_p,\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ *\overline{k}_p,\ \mathbf{val}\ Q'\,]$ for some Q', or

2. $[\,\vec{\mathcal{S}},\ *\overline{k}_p,\ M'[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \overset{*}{\underset{l+1}{\Rightarrow}} [\,\vec{\mathcal{S}},\ *\overline{k}_p,\ \ldots,\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\,]$.

In the first case, since $k_j < l+1$ for any j, we get $M'[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \overline{Q}'$ by the induction hypothesis. Therefore,

$$M[\overline{\vec{R}}/\vec{x}] \quad = \quad M'[\overline{\vec{R}}/\vec{x}]\,u_p \quad \overset{*}{\underset{\text{CBV}}{\longrightarrow}} \quad \overline{Q}'\,u_p.$$

On the other hand, $[\,\vec{\mathcal{S}},\ *\overline{k}_p,\ \mathbf{val}\ Q'\,] \overset{*}{\underset{l}{\Rightarrow}} [\,\vec{\mathcal{S}},\ \ldots,\ \mathbf{throw}\ \overline{q}\ *,\ \mathbf{val}\ Q\,]$ by (3.2). Therefore, we get $Q' = \kappa\,w.\,\mathbf{throw}\ w\ (\mathbf{val}\ Q)$ for some w and $q = k_p$. That is, $\overline{Q}' = \kappa\,w.\,\mathbf{throw}\ w\ \overline{Q}$ and

$$\begin{array}{rcl} M[\overline{\vec{R}}/\vec{x}] & \overset{*}{\underset{\text{CBV}}{\longrightarrow}} & \overline{Q}'\,u_p \\ & = & (\kappa\,w.\,\mathbf{throw}\ w\ \overline{Q})\,u_p \\ & \underset{\text{CBV}}{\succ} & \mathbf{throw}\ u_p\ \overline{Q}. \end{array}$$

In the second case, we get $M'[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}]$ for some j and $\mathcal{C}$ such that $q = k_j$ and $\mathcal{C}'$ does not capture u_j, by the induction hypothesis. Therefore,

$$\begin{array}{rcl} M[\overline{\vec{R}}/\vec{x}] & = & M'[\overline{\vec{R}}/\vec{x}]\,u_p \\ & \underset{\text{CBV}}{\succ} & \mathcal{C}'[\mathbf{throw}\ u_j\ \overline{Q}]\,u_p \\ & = & \mathcal{C}[\mathbf{throw}\ u_j\ \overline{Q}], \end{array}$$

where $\mathcal{C} = \mathcal{C}'\,u_p$, which does not capture u_j.

Case 8: *$M = \lambda\,y.\,M'$ for some y and M'.* Note that $M[\overline{\vec{R}}/\vec{x}] \underset{\text{CBV}}{\succ} \lambda\,y.\,(M'[\overline{\vec{R}}/\vec{x}])$. Since $FTV(\lambda\,x.\,M') = \{\}$, we get $M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}] = \lambda\,y.\,(M[\mathbf{val}\,\vec{R}/\vec{x}])$. Therefore, by the definition of transition rules,

$$[\,\vec{\mathcal{S}},\ M[\mathbf{val}\,\vec{R}/\vec{x}, \overline{\vec{k}}/\vec{u}]\,] \underset{l}{\Rightarrow} [\,\vec{\mathcal{S}},\ \mathbf{val}\ (\lambda\,y.\,(M'[\mathbf{val}\,\vec{R}/\vec{x}]))\,].$$

So (3.2) does not hold. Suppose (3.1), i.e., $Q = \lambda\,y.\,(M'[\mathbf{val}\,\vec{R}/\vec{x}])$. We get $\overline{Q} = \lambda\,y.\,(M'[\overline{\vec{R}}/\vec{x}])$ by Proposition 3.1.8.

Case 9: *M has one of other forms.* Similar. □

Corollary 3.2.10 *Let M be a closed term. Let V and Q be a value and an internal value, respectively.*

1. $M \underset{\text{CBV}}{\succ} V$ implies $[\,M\,] \overset{*}{\Rightarrow} [\,\mathbf{val}\ Q\,]$ for some Q such that $\overline{Q} = V$, and

2. $[\,M\,] \overset{*}{\Rightarrow} [\,\mathbf{val}\ Q\,]$ implies $M \underset{\text{CBV}}{\succ} \overline{Q}$.

Proof. Straightforward from Theorem 3.2.4 and Theorem 3.2.9. □

3.3 Realizability by the abstract machine

We can give another realizability interpretation of the formal system in terms of the abstract machine.

Theorem 3.3.1 *The relation*

$$\{x_1 : A_1, \ldots, x_m : A_m\} \models M : C \,;\, \{u_1 : B_1, \ldots, u_n : B_n\}$$

holds if and only if for any closed terms $K_1, \ldots, K_m$ *such that* $[\ K_i\ ] \overset{*}{\Rightarrow} [\ \mathbf{val}\ R_i\ ]$ *and* $\overline{R_i}\ \mathbf{r}\ A_i$ *for some* R_i *(*$1 \leq i \leq m$*), for any context segments* $\mathcal{S}_1, \ldots, \mathcal{S}_l$*, and for any tag constants* $\overline{k}_1, \ldots, \overline{k}_n$ *such that* $k_j < l$ *for any* j*,*

1. $[\ \mathcal{S}_1,\ \ldots,\ \mathcal{S}_l,\ M[\vec{K}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ] \underset{l}{\overset{*}{\Rightarrow}} [\ \mathcal{S}_1,\ \ldots,\ \mathcal{S}_l,\ \mathbf{val}\ Q\ ]$ *and* $\overline{Q}\ \mathbf{r}\ C$ *for some* Q*, or*
2. $[\ \mathcal{S}_1,\ \ldots,\ \mathcal{S}_l,\ M[\vec{K}/\vec{x}, \vec{\overline{k}}/\vec{u}]\ ] \underset{l}{\overset{*}{\Rightarrow}} [\ \mathcal{S}_1,\ \ldots,\ \mathcal{S}_l,\ \ldots,\ \mathbf{throw}\ \overline{k}_j\ *,\ \mathbf{val}\ Q]$ *and* $\overline{Q}\ \mathbf{r}\ B_j$ *for some* j *and* Q*,*

where the term $M[\vec{K}/\vec{x}, \vec{\overline{k}}/\vec{u}]$ *stands for* $M[K_1/x_1, \ldots, K_m/x_m, \overline{k}_1/u_1, \ldots, \overline{k}_n/u_n]$*.*

Proof. Straightforward from Theorem 3.2.4 and Theorem 3.2.9. □

Chapter 4

The typing system as a logic

In this chapter we discuss $L_{c/t}^{\text{CBV}}$ considering it as a logic. We reformulate the typing system $L_{c/t}^{\text{CBV}}$ into a sequent calculus since the logic can be easily understood when compared with LK and LJ.

4.1 A sequent calculus style formulation

We consider types as *formulas*. That is, we have atomic formulas, conjunctions, disjunctions, implications and exceptions. A *sequent* of the system is of the form

$$A_1 \dots A_m \to C\,;\; E_1 \dots E_n$$

where m and n can be 0. It looks like a sequent of LK rather than LJ ignoring the semicolon ";" between C and $E_1 \dots E_n$. Actually, its purely logical meaning is the same as LK. In this sense, the semicolon is unnecessary. But we saw that it plays a significant role for the constructive meaning of the sequent.

Definition 4.1.1 (Realizability of sequents) The sequents are interpreted as follows. Let $<x_1 \dots x_m,\ M,\ u_1 \dots u_n>$ be a triple which consists of a sequence of distinct individual variables $x_1 \dots x_m$, a term M and a sequence of distinct tag variables $u_1 \dots u_n$. We assume that the free individual and tag variables of the term M are included in the two sequences. A triple $<x_1 \dots x_m,\ M,\ u_1 \dots u_n>$ realizes $A_1 \dots A_m \to C\,;\; B_1 \dots B_n$ if and only if

$$\{x_1 : A_1, \dots, x_m : A_m\} \models M : C\,;\; \{u_1 : B_1, \dots, u_n : B_n\}.$$

Definition 4.1.2 (Inference rules) The inference rules and the corresponding realizers are as follows.

$$\dfrac{}{\begin{array}{c} A \to A\,; \\ <x,\ x,\ > \end{array}}\ (init) \qquad \dfrac{\begin{array}{cc} <\vec{x},\ M_1,\ \vec{u}> & <\vec{y}\,z,\ M_2,\ \vec{v}> \\ \Gamma_1 \to A\,;\ \Delta_1 & \Gamma_2\, A \to C\,;\ \Delta_2 \end{array}}{\begin{array}{c} \Gamma_1\, \Gamma_2 \to C\,;\ \Delta_1\, \Delta_2 \\ <\vec{x}\,\vec{y},\ \mathbf{let}\ z{=}M_1.\ M_2,\ \vec{u}\,\vec{v}> \end{array}}\ (cut)$$

$$\dfrac{\begin{array}{c} <\vec{x}\,y_1\,y_2\,\vec{z},\ M,\ \vec{u}> \\ \Gamma_1\, A\, B\, \Gamma_2 \to C\,;\ \Delta \end{array}}{\begin{array}{c} \Gamma_1\, B\, A\, \Gamma_2 \to C\,;\ \Delta \\ <\vec{x}\,y_2\,y_1\,\vec{z},\ M,\ \vec{u}> \end{array}}\ (x{\to}) \qquad \dfrac{\begin{array}{c} <\vec{x},\ M,\ \vec{u}> \\ \Gamma \to C\,;\ \Delta \end{array}}{\begin{array}{c} \Gamma\, A \to C\,;\ \Delta \\ <\vec{x}\,y,\ M,\ \vec{u}> \end{array}}\ (w{\to}) \qquad \dfrac{\begin{array}{c} <\vec{x}\,y_1\,y_2\,\vec{z},\ M,\ \vec{u}> \\ \Gamma_1\, A\, A\, \Gamma_2 \to C\,;\ \Delta \end{array}}{\begin{array}{c} \Gamma_1\, A\, \Gamma_2 \to C\,;\ \Delta \\ <\vec{x}\,y\,\vec{z},\ M[y/y_1, y/y_2],\ \vec{u}> \end{array}}\ (c{\to})$$

$$\dfrac{\begin{array}{c}<\vec{x},\ M,\ \vec{u}\,v_1\,v_2\,\vec{w}>\\ \Gamma \to A\,;\ \Delta_1\,B\,C\,\Delta_2\end{array}}{\begin{array}{c}\Gamma \to A\,;\ \Delta_1\,C\,B\,\Delta_2\\ <\vec{x},\ M,\ \vec{u}\,v_2\,v_1\,\vec{w}>\end{array}}\,(\to x) \qquad \dfrac{\begin{array}{c}<\vec{x},\ M,\ \vec{v}>\\ \Gamma \to A\,;\ \Delta\end{array}}{\begin{array}{c}\Gamma \to A\,;\ E\,\Delta\\ <\vec{x},\ M,\ u\,\vec{v}>\end{array}}\,(\to w) \qquad \dfrac{\begin{array}{c}<\vec{x},\ M,\ u_1\,u_2\,\vec{v}>\\ \Gamma \to A\,;\ E\,E\,\Delta\end{array}}{\begin{array}{c}\Gamma \to A\,;\ E\,\Delta\\ <\vec{x},\ M[u/u_1, u/u_2],\ u\,\vec{v}>\end{array}}\,(\to c)$$

$$\dfrac{\begin{array}{c}<\vec{x},\ M,\ \vec{v}>\\ \Gamma \to E\,;\ \Delta\end{array}}{\begin{array}{c}\Gamma \to A\,;\ E\,\Delta\\ <\vec{x},\ \mathbf{throw}\ u\ M,\ u\,\vec{v}>\end{array}}\,(throw) \qquad \dfrac{\begin{array}{c}<\vec{x},\ M,\ u\,\vec{v}>\\ \Gamma \to A\,;\ A\,\Delta\end{array}}{\begin{array}{c}\Gamma \to A\,;\ \Delta\\ <\vec{x},\ \mathbf{catch}\ u\ M,\ \vec{v}>\end{array}}\,(catch)$$

$$\dfrac{\begin{array}{c}<\vec{x}\,z,\ M,\ \vec{u}>\\ \Gamma\,A \to C\,;\ \Delta\end{array}}{\begin{array}{c}\Gamma\,A \wedge B \to C\,;\ \Delta\\ <\vec{x}\,y,\ M[\mathbf{proj_1}\,y/z],\ \vec{u}>\end{array}}\,(\wedge_1 \to) \qquad \dfrac{\begin{array}{c}<\vec{x}\,z,\ M,\ \vec{u}>\\ \Gamma\,B \to C\,;\ \Delta\end{array}}{\begin{array}{c}\Gamma\,A \wedge B \to C\,;\ \Delta\\ <\vec{x}\,y,\ M[\mathbf{proj_2}\,y/z],\ \vec{u}>\end{array}}\,(\wedge_2 \to)$$

$$\dfrac{\begin{array}{cc}<\vec{x},\ M_1,\ \vec{u}> & <\vec{y},\ M_2,\ \vec{v}>\\ \Gamma_1 \to A\,;\ \Delta_1 & \Gamma_2 \to B\,;\ \Delta_2\end{array}}{\begin{array}{c}\Gamma_1\,\Gamma_2 \to A \wedge B\,;\ \Delta_1\,\Delta_2\\ <\vec{x}\,\vec{y},\ <M_1,\ M_2>,\ \vec{u}\,\vec{v}>\end{array}}\,(\to \wedge) \qquad \dfrac{\begin{array}{cc}<\vec{x}\,z_1,\ M_1,\ \vec{u}> & <\vec{y}\,z_2,\ M_2,\ \vec{v}>\\ \Gamma_1\,A \to C\,;\ \Delta_1 & \Gamma_2\,B \to C\,;\ \Delta_2\end{array}}{\begin{array}{c}\Gamma_1\,\Gamma_2\,A \vee B \to C\,;\ \Delta_1\,\Delta_2\\ <\vec{x}\,\vec{y}\,z,\ \mathbf{case}\ z\ z_1.M_1\ z_2.M_2,\ \vec{u}\,\vec{v}>\end{array}}\,(\vee \to)$$

$$\dfrac{\begin{array}{c}<\vec{x},\ M,\ \vec{u}>\\ \Gamma \to A\,;\ \Delta\end{array}}{\begin{array}{c}\Gamma \to A \vee B\,;\ \Delta\\ <\vec{x},\ \mathbf{inj_1}\,M,\ \vec{u}>\end{array}}\,(\to \vee_1) \qquad \dfrac{\begin{array}{c}<\vec{x},\ M,\ \vec{u}>\\ \Gamma \to B\,;\ \Delta\end{array}}{\begin{array}{c}\Gamma \to A \vee B\,;\ \Delta\\ <\vec{x},\ \mathbf{inj_2}\,M,\ \vec{u}>\end{array}}\,(\to \vee_2)$$

$$\dfrac{\begin{array}{cc}<\vec{x},\ M_1,\ \vec{u}> & <\vec{y}\,z',\ M_2,\ \vec{v}>\\ \Gamma_1 \to A\,;\ \Delta_1 & \Gamma_2\,B \to C\,;\ \Delta_2\end{array}}{\begin{array}{c}\Gamma_1\,\Gamma_2\,A \supset B \to C\,;\ \Delta_1\,\Delta_2\\ <\vec{x}\,\vec{y}\,z,\ \mathbf{let}\ z'{=}zM_1\,.\ M_2,\ \vec{u}\,\vec{v}>\end{array}}\,(\supset \to) \qquad \dfrac{\begin{array}{c}<\vec{x}\,y,\ M,\ >\\ \Gamma\,A \to B\,;\end{array}}{\begin{array}{c}\Gamma \to A \supset B\,;\\ <\vec{x},\ \lambda y.\,M,\ >\end{array}}\,(\to \supset)$$

$$\dfrac{\begin{array}{cc}<\vec{x}\,z_1,\ M_1,\ \vec{v}> & <\vec{y}\,z_2,\ M_2,\ \vec{w}>\\ \Gamma_1\,A \to B\,;\ \Delta_1 & \Gamma_2\,E \to C\,;\ \Delta_2\end{array}}{\begin{array}{c}\Gamma_1\,\Gamma_2\,A \triangleleft E \to C\,;\ B\,\Delta_1\,\Delta_2\\ <\vec{x}\,\vec{y}\,z,\ \mathbf{let}\ z_2{=}(\mathbf{catch}\ u'\ \mathbf{let}\ z_1{=}zu'.\ \mathbf{throw}\ u\ M_1).\ M_2,\ u\,\vec{v}\,\vec{w}>\end{array}}\,(\triangleleft \to)$$

$$\dfrac{\begin{array}{c}<\vec{x},\ M,\ u\,\vec{v}>\\ \Gamma \to A\,;\ E\,\Delta\end{array}}{\begin{array}{c}\Gamma \to A \triangleleft E\,;\ \Delta\\ <\vec{x},\ \kappa u.\,M,\ \vec{v}>\end{array}}\,(\to \triangleleft)$$

The typing system $L^{\text{CBV}}_{c/t}$ is equivalent to the above sequent calculus as a logic.

Theorem 4.1.3 $\{x_1 : A_1, \ldots, x_m : A_m\} \vdash M : C\,;\ \{u_1 : B_1, \ldots, u_n : B_n\}$ *is derivable in* $L^{\text{CBV}}_{c/t}$ *for some* $x_1, \ldots, x_m$*,* M*,* $u_1, \ldots, u_n$*, if and only if* $A_1 \ldots A_m \to C\,;\ B_1 \ldots B_n$ *is derivable in the sequent calculus.*

Proof. Induction on the structure of the derivation. □

If we ignore semicolons in the sequents, the inference rules are almost the same as the ones of LK. It should be noted that every right logical rule introduces a logical connective into the formula between the arrow and the semicolon, namely, the main conclusion. In the following sections we discuss the details.

4.2 The logical meaning of the new connective

First we consider the new logical connective $\triangleleft$. From the logical point of view, the new connective $\triangleleft$ is equivalent to $\vee$. The difference between them consists only in their implementations.

Definition 4.2.1 We use $\tilde{A}$ to denote the formula obtained from a formula A by replacing every occurrence of the logical connective $\triangleleft$ by $\vee$. If $\tilde{A} = \tilde{B}$, then we denote it by $A \simeq B$. If $\Gamma = A_0 \ldots A_n$, $\Delta = B_0 \ldots B_n$ and $A_i \simeq B_i$ for any i $(0 \leq i \leq n)$, then we denote it by $\Gamma \simeq \Delta$.

Lemma 4.2.2 *If $A \simeq A'$, then $A \rightarrow A'\,;$ is a derivable sequent.*

Proof. Straightforward induction on the structure of the formula A. The basic idea comes from the following two derivations.

$$
\dfrac{\dfrac{\dfrac{\vdots\ \text{ind. hyp.}}{A \rightarrow A'\,;}}{A \rightarrow A' \vee B'\,;}\,(\rightarrow\vee_1) \quad \dfrac{\dfrac{\vdots\ \text{ind. hyp.}}{B \rightarrow B'\,;}}{B \rightarrow A' \vee B'\,;}\,(\rightarrow\vee_2)}{\dfrac{A \triangleleft B \rightarrow A' \vee B'\,;\, A' \vee B'}{A \triangleleft B \rightarrow A' \vee B'\,;}\,(catch)}\,(\triangleleft\rightarrow)
$$

$$
\dfrac{\dfrac{\vdots\ \text{ind. hyp.}}{A \rightarrow A'\,;} \quad \dfrac{\dfrac{\vdots\ \text{ind. hyp.}}{B \rightarrow B'\,;}}{B \rightarrow A'\,;\, B'}\,(throw)}{\dfrac{A \vee B \rightarrow A'\,;\, B'}{A \vee B \rightarrow A' \triangleleft B'\,;}\,(\rightarrow\triangleleft)}\,(\vee\rightarrow)
$$

□

Theorem 4.2.3 *If $\Gamma \rightarrow A\,;\,\Delta$ is a derivable sequent. and if $\Gamma \simeq \Gamma'$, $A \simeq A'$ and $\Delta \simeq \Delta'$, then $\Gamma' \rightarrow A'\,;\,\Delta'$ is also derivable.*

Proof. Induction on the structure of the derivation of the sequent $\Gamma \rightarrow A\,;\Delta$. Apply Lemma 4.2.2 in the case that the last rule is $(init)$. □

If we identify $A \triangleleft B$ with $A \vee B$, the formal system can be regarded as a variant of the propositional fragment of LJ.

Theorem 4.2.4 *A sequent $A_1 \ldots A_m \rightarrow C\,;$ is derivable in our sequent calculus if and only if $\tilde{A}_1 \ldots \tilde{A}_m \rightarrow \tilde{C}$ is derivable in (the propositional fragment of) LJ.*

Proof. The *if* part is trivial because the propositional fragment of LJ can be regarded as a subsystem of ours. For the *only if* part, prove the following lemma by induction on the structure of the derivation: *If $A_1 \ldots A_m \rightarrow C\,; E_1 \ldots E_n$ is derivable, then $\tilde{A}_1 \ldots \tilde{A}_m \rightarrow \tilde{C} \vee \tilde{E}_1 \vee \ldots \vee \tilde{E}_n$ is derivable in (the propositional fragment of) LJ.* The theorem is a corollary of the lemma. □

As a corollary of the theorem, we get the disjunction property of the system.

Corollary 4.2.5 (Disjunction property) *Let $\rightarrow A \vee B\,;$ be a derivable sequent. Then we can derive $\rightarrow A\,;$ or $\rightarrow B\,;$.*

We should note that there is another possibility of formulation of $(\triangleleft\!\to)$ as follows.

$$\dfrac{\begin{array}{c}<\vec{x}\,z,\ M,\ \vec{v}>\\ \Gamma\, A \to C\,;\,\Delta\end{array}}{\begin{array}{c}\Gamma\, A \triangleleft E \to C\,;\,E\,\Delta\\ <\vec{x}\,y,\ \mathbf{let}\ z{=}y\,u.\ M,\ u\,\vec{v}>\end{array}}\ (\triangleleft\!\to)$$

This formulation seems more natural than the original in Definition 4.1.2 from the viewpoint of realizer construction, and is equivalent to the original as long as we have (cut). But we could not get the cut-elimination theorem if we replaced the original rule by this rule.

4.3 The catch and throw mechanism as structural rules

The three right-structural rules of LK are divided into five rules, i.e., $(\to x)$, $(\to c)$, $(\to w)$, $(catch)$ and $(throw)$. The rules $(catch)$ and $(\to c)$ correspond to the right-contraction rule of LK. The former introduces a catch-term. The latter means a sharing of one tag variable by two throw-terms, i.e., multiple throw-terms can be caught by one catch-term afterwards. The right-weakening rule of LK is divided into $(throw)$ and $(\to w)$. The former corresponds to a throw-term. The latter means an introduction of a redundant tag variable. We note that the rule $(\to w)$ is a derived rule.

$$<\vec{x},\ M,\ \vec{v}>$$
$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \to A\,;\,\Delta}{\Gamma \to E\,;\,A\,\Delta}\,(throw)}{\Gamma \to A\,;\,E\,A\,\Delta}\,(throw)}{\Gamma \to A\,;\,A\,E\,\Delta}\,(\to x)}{\Gamma \to A\,;\,E\,\Delta}\,(catch)$$
$$<\vec{x},\ \mathbf{catch}\ w\ (\mathbf{throw}\ w\ (\mathbf{throw}\ u\ M)),\ u\,\vec{v}>$$

But we adopt $(\to w)$ as a primitive rule because the realizer given above introduces a redundant throw-term, i.e., a throw-term never invoked.

We have no right-exchange rule over the semicolon, but it is also a derived rule as follows.

$$<\vec{x},\ M,\ u\,\vec{w}>$$
$$\dfrac{\dfrac{\dfrac{\Gamma \to A\,;\,E\,\Delta}{\Gamma \to E\,;\,A\,E\,\Delta}\,(throw)}{\Gamma \to E\,;\,E\,A\,\Delta}\,(\to x)}{\Gamma \to E\,;\,A\,\Delta}\,(catch)$$
$$<\vec{x},\ \mathbf{catch}\ u\ (\mathbf{throw}\ v\ M),\ v\,\vec{w}>$$

In contrast to $(\to w)$, we leave it as a derived rule because there is no primitive programming construct corresponding to the rule.

4.4 The restriction on the right implication rule

As a logic, the only and significant difference from LK is that there must be exactly one formula on the right hand side of the sequent when we apply the right-implication rule $(\to\supset)$. This restriction is required to keep the system constructive. Roughly, our system can be regarded as the propositional fragment of LK with this restriction on the right-implication rule. If we dropped the restriction, the following anomaly would occur. Consider the following derivation of $A \vee (A \supset B)$, which is not derivable in the constructive logic.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \rightarrow A\,;}\,(init)}{A \rightarrow A \vee (A \supset B)\,;}\,(\rightarrow\vee_1)}{A \rightarrow B\,;\ A \vee (A \supset B)}\,(throw)}{\rightarrow A \supset B\,;\ A \vee (A \supset B)}\,(\rightarrow\supset)}{\rightarrow A \vee (A \supset B)\,;\ A \vee (A \supset B)}\,(\rightarrow\vee_2)}{\rightarrow A \vee (A \supset B)\,;}\,(catch)
$$

The realizer would be $<,\ \mathbf{catch}\ u\ (\mathbf{inj_2}\,(\lambda\, x.\,\mathbf{throw}\ u\ (\mathbf{inj_1}\ x))),\ >$. Note that the term is a normal form since $\lambda\, x.\,\mathbf{throw}\ u\ (\mathbf{inj_1}\ x)$ is not a value. But it does not realize $A \vee (A \supset B)$. The evaluation process of the term by the abstract machine would be as follows.

$$
\begin{aligned}
&[\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_{l-1},\ \mathbf{case} * y.e_1\ z.e_2,\ \mathbf{catch}\ u\ (\mathbf{inj_2}\,(\lambda\, x.\,\mathbf{throw}\ u\ (\mathbf{inj_1}\ x)))\,] \\
&\qquad \Rightarrow\ [\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_{l-1},\ \mathbf{case} * y.e_1\ z.e_2,\ \mathbf{inj_2}\,(\lambda\, x.\,\mathbf{throw}\ \bar{l}\ (\mathbf{inj_1}\ x))\,] \\
&\qquad \Rightarrow\ [\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_{l-1},\ \mathbf{case} * y.e_1\ z.e_2,\ \mathbf{inj_2} *,\ \lambda\, x.\,\mathbf{throw}\ \bar{l}\ (\mathbf{inj_1}\ x)\,] \\
&\qquad \Rightarrow\ [\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_{l-1},\ \mathbf{case} * y.e_1\ z.e_2,\ \mathbf{inj_2} *,\ \mathbf{val}\ (\lambda\, x.\,\mathbf{throw}\ \bar{l}\ (\mathbf{inj_1}\ x))\,] \\
&\qquad \Rightarrow\ [\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_{l-1},\ \mathbf{case} * y.e_1\ z.e_2,\ \mathbf{val}\ (\mathbf{inj_2}\,(\lambda\, x.\,\mathbf{throw}\ \bar{l}\ (\mathbf{inj_1}\ x)))\,] \\
&\qquad \Rightarrow\ [\,\mathcal{S}_1,\ \ldots,\ \mathcal{S}_{l-1},\ e_2[\mathbf{val}\ (\lambda\, x.\,\mathbf{throw}\ \bar{l}\ (\mathbf{inj_1}\ x))/z]\,]
\end{aligned}
$$

Note that the tag constant $\bar{l}$ in the last is meaningless because the corresponding evaluation context has been lost. From a computational point of view, this problem can be solved by introducing more powerful facilities for non-local exit such as *call/cc* of Scheme. But it affects the realizability interpretation of formulas. For example, although the realizers of disjunctions still have a certain constructive meaning, they do not always contain the information that specifies which of $A \vee B$ is realized by them. It should be noted that the system without the restriction becomes a classical one, and we do not have the soundness theorem or the disjunction property anymore.

LK with this kind of restriction on $(\rightarrow\supset)$ is known as a variant of LJ, which is essentially equivalent to LJ (cf.[18, 33, 34]). The same restriction is also required for $(\rightarrow\forall)$-rule in the case of predicate calculus.

4.5 The cut-elimination theorem

The (cut)-rule of our sequent calculus is redundant.

Theorem 4.5.1 (Cut-elimination) *If a sequent is derivable, then we can derive it without (cut)-rule.*

The proof becomes more complicated than the case of LJ/LK because there exists the special restriction to apply $(\rightarrow\supset)$-rule and the new connective $\triangleleft$ has been introduced. We call the following inference rules *structural rules.*

$$(x\rightarrow)\quad (c\rightarrow)\quad (w\rightarrow)\quad (\rightarrow x)\quad (\rightarrow c)\quad (\rightarrow w)\quad (throw)\quad (catch)$$

Note that we count (*throw*) and (*catch*) as structural rules. *Left-logical rules, right-logical rules* and *principal formulas* of logical rules are defined in the standard manner. To prove Theorem 4.5.1, we first extend (*cut*)-rule as follows, and we call the extended rule (*mix*).

$$\frac{\Gamma_1 \rightarrow A\,;\ \Delta_1 \qquad \Gamma_2 \rightarrow C\,;\ \Delta_2}{\Gamma_1\,(\Gamma_2 - X) \rightarrow C\,;\ (A\,\Delta_1 - X)\,\Delta_2}\ (mix)$$

where $\Gamma_1 - X$ stands for the sequence of formulas obtained from Γ_1 by removing X, and $A\,\Delta_1 - X$ stands for the sequence of formulas obtained from $A\,\Delta_1$ by removing X.

We now consider only derivations that involve (*mix*) instead of (*cut*) since we can use (*mix*) instead of (*cut*). It is enough to show that we can construct a derivation which does not include any (*mix*), i.e., a mix-free derivation. We prove the theorem by induction on the number of occurrences of (*mix*) in the derivation. Suppose we have a derivation Π such as

$$\frac{\dfrac{\vdots\ \Pi_1}{\Gamma_1 \rightarrow A\,;\ \Delta_1}\ (r1) \qquad \dfrac{\vdots\ \Pi_2}{\Gamma_2 \rightarrow C\,;\ \Delta_2}\ (r2)}{\Gamma_1\,(\Gamma_2 - X) \rightarrow C\,;\ (A\,\Delta_1 - X)\,\Delta_2}\ (mix)$$

where Π_1 and Π_2 are mix-free derivations and $(r1)$ and $(r2)$ stand for some inference rules except (*mix*). We translate derivations of this form to mix-free derivations.

Definition 4.5.2 (Grades and heights) We define the grade of the (*mix*)-rule as the number of logical connectives such as $\wedge, \vee, \supset$ and $\triangleleft$ occurring in X. The left height of the (*mix*)-rule is the maximum length of the derivation paths in Π_1. The right height of the (*mix*)-rule is the maximum length of the derivation paths in Π_2. The height of the (*mix*)-rule is the sum of the left and right heights.

Proof of the cut-elimination theorem. We translate derivations of the form of Π to mix-free derivations by induction on the lexicographic ordering of the grade and the height of the (*mix*)-rule. We divide the proof into the following four cases.

Case 1: *The rule $(r1)$ is not a right-logical rule whose principal formula is X.* The only crucial is the subcase that $(r1)$ is $(\rightarrow\supset)$ whose principal formula is not X. Since there exists a restriction to apply $(\rightarrow\supset)$, this case is not just a variant of other subcases. The derivation Π has the following form.

$$\frac{\dfrac{\begin{array}{c}\vdots\ \Pi_1\\ \Gamma_1\,A \rightarrow B\,;\end{array}}{\Gamma_1 \rightarrow A \supset B\,;}\ (\rightarrow\supset) \qquad \dfrac{\vdots\ \Pi_2}{\Gamma_2 \rightarrow C\,;\ \Delta_2}\ (r2)}{\Gamma_1\,(\Gamma_2 - X) \rightarrow C\,;\ (A \supset B - X)\,\Delta_2}\ (mix)$$

The conclusion is identical to $\Gamma_1\,(\Gamma_2 - X) \rightarrow C\,;\ A \supset B\,\Delta_2$ since the principal formula $A \supset B$ is not X. We can derive it applying structural rules to $\Gamma_1 \rightarrow A \supset B\,;$.

Case 2: *The rule $(r1)$ is a right-logical rule whose principal formula is X, and $(r2)$ is $(\rightarrow\supset)$.* This case is crucial.

Subcase 2-1: The rule $(r1)$ *is* $(\to\wedge)$ *whose principal formula is* X*, i.e., the derivation* Π *has the following form.*

$$\dfrac{\dfrac{\dfrac{\vdots\ \Pi_{11}}{\Gamma_{11}\to A_1\,;\,\Delta_{11}} \quad \dfrac{\vdots\ \Pi_{12}}{\Gamma_{12}\to A_2\,;\,\Delta_{12}}}{\Gamma_{11}\,\Gamma_{12}\to A_1\wedge A_2\,;\,\Delta_{11}\,\Delta_{12}}(\to\wedge) \quad \dfrac{\dfrac{\vdots\ \Pi_2}{\Gamma_2\,C\to D\,;}}{\Gamma_2\to C\supset D\,;}(\to\supset)}{\Gamma_{11}\,\Gamma_{12}\,(\Gamma_2-X)\to C\supset D\,;\,(\Delta_{11}\,\Delta_{12}-X)}(mix)$$

First, we consider the following derivation for each i $(i=1,2)$.

$$\dfrac{\dfrac{\vdots\ \Pi_{1i}}{\Gamma_{1i}\to A_i\,;\,\Delta_{1i}} \quad \dfrac{\dfrac{\vdots\ \Pi_2}{\Gamma_2\,C\to D\,;}}{\Gamma_2\to C\supset D\,;}(\to\supset)}{\Gamma_{1i}\,(\Gamma_2-X)\to C\supset D\,;\,(A_i\,\Delta_{1i}-X)}(mix)$$

Since the left height of the (mix)-rule is less than Π, we have a mix-free form of this derivation by the induction hypothesis. Let Π'_{1i} be the mix-free derivation. Next, we consider the following derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A_1\to A_1\,;}(init) \quad \dfrac{}{A_2\to A_2\,;}(init)}{A_1\,A_2\to A_1\wedge A_2\,;}(\to\wedge) \quad \dfrac{\vdots\ \Pi_2}{\Gamma_2\,C\to D\,;}}{\dfrac{A_1\,A_2\,(\Gamma_2\,C-X)\to D\,;}{A_1\,A_2\,(\Gamma_2-X)\,C\to D\,;}(w\!\to)^*}(mix)}{A_1\,A_2\,(\Gamma_2-X)\to C\supset D\,;}(\to\supset)$$

Since the left height of the (mix)-rule is equal to or less than Π and the right height is less than Π, we have a mix-free form of this derivation by the induction hypothesis. Let Π'_2 be the mix-free derivation. Combining Π'_{11}, Π'_{12} and Π'_2, we get the following derivation, where Γ'_2 and Δ'_{1i} are Γ_2-X and $A_i\,\Delta_{1i}-X$, respectively.

$$\dfrac{\dfrac{\vdots\ \Pi'_{11}}{\Gamma_{11}\,\Gamma'_2\to C\supset D\,;\,\Delta'_{11}} \quad \dfrac{\vdots\ \Pi'_2}{A_1\,A_2\,\Gamma'_2\to C\supset D\,;}}{\underline{\Gamma_{11}\,\Gamma'_2\,(A_1\,A_2\,\Gamma'_2-A_1)\to C\supset D\,;\,(C\supset D\,\Delta'_{11}-A_1)}}(mix)$$

$$\dfrac{\dfrac{\dfrac{\vdots\ \Pi'_{12}}{\Gamma_{12}\,\Gamma'_2\to C\supset D\,;\,\Delta'_{12}} \quad \dfrac{\vdots\ \text{structural rules}}{A_2\,\Gamma_{11}\,\Gamma'_2\to C\supset D\,;\,\Delta_{11}-X}}{\Gamma_{12}\,\Gamma'_2\,(A_2\,\Gamma_{11}\,\Gamma'_2-A_2)\to C\supset D\,;\,(C\supset D\,\Delta'_{12}-A_2)\,(\Delta_{11}-X)}(mix)}{\dfrac{\vdots\ \text{structural rules}}{\Gamma_{11}\,\Gamma_{12}\,\Gamma'_2\to C\supset D\,;\,(\Delta_{11}\,\Delta_{12}-X)}}$$

Since the grades of the two (mix)-rules are both less than Π, we have a mix-free form of this derivation by the induction hypothesis.

Subcase 2-2: The rule $(r1)$ *is* $(\to\vee_1)$ *or* $(\to\vee_2)$ *whose principal formula is* X*.* Similar to Case 2-1.

Subcase 2-3: The rule (r1) is $(\to\supset)$ *whose principal formula is* X. In this case, just similar to the proof for LK or LJ because Δ_1 is empty.

Subcase 2-4: The rule (r1) is $(\to\lhd)$ *whose principal formula is* X*, i.e., the derivation* Π *has the following form.*

$$\dfrac{\dfrac{\begin{matrix}\vdots\ \Pi_1\\ \Gamma_1 \to A_1\,;\ A_2\,\Delta_1\end{matrix}}{\Gamma_1 \to A_1 \lhd A_2\,;\ \Delta_1}(\to\lhd) \qquad \dfrac{\begin{matrix}\vdots\ \Pi_2\\ \Gamma_2\,C \to D\,;\end{matrix}}{\Gamma_2 \to C \supset D\,;}(\to\supset)}{\Gamma_1\,(\Gamma_2 - X) \to C \supset D\,;\ (\Delta_1 - X)}(mix)$$

First, We consider the following derivation.

$$\dfrac{\begin{matrix}\vdots\ \Pi_1\\ \Gamma_1 \to A_1\,;\ A_2\,\Delta_1\end{matrix} \qquad \dfrac{\begin{matrix}\vdots\ \Pi_2\\ \Gamma_2\,C \to D\,;\end{matrix}}{\Gamma_2 \to C \supset D\,;}(\to\supset)}{\Gamma_1\,(\Gamma_2 - X) \to C \supset D\,;\ A_1\,A_2\,\Delta_1 - X}(mix)$$

Since the left height of the (mix)-rule is less than Π, we have a mix-free form of this derivation by the induction hypothesis. Let Π'_1 be the mix-free derivation. Next, we consider the following two derivations.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A_1 \to A_1\,;}(init)}{A_1 \to A_1\,;\ A_2}(\to w)}{A_1 \to A_1 \lhd A_2\,;}(\to\lhd) \qquad \begin{matrix}\vdots\ \Pi_2\\ \Gamma_2\,C \to D\,;\end{matrix}}{A_1\,(\Gamma_2\,C - X) \to D\,;}(mix)}{A_1\,(\Gamma_2 - X)\,C \to D\,;}(w\to)^*}{A_1\,(\Gamma_2 - X) \to C \supset D\,;}(\to\supset)
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A_2 \to A_2\,;}(init)}{A_2 \to A_1\,;\ A_2}(throw)}{A_2 \to A_1 \lhd A_2\,;}(\to\lhd) \qquad \begin{matrix}\vdots\ \Pi_2\\ \Gamma_2\,C \to D\,;\end{matrix}}{A_2\,(\Gamma_2\,C - X) \to D\,;}(mix)}{A_2\,(\Gamma_2 - X)\,C \to D\,;}(w\to)^*}{A_2\,(\Gamma_2 - X) \to C \supset D\,;}(\to\supset)$$

Note that we need at least two steps to derive $\Gamma_1 \to A_1\,;\ A_2$ in Π_1. Therefore the left heights of the (mix)-rules are equal to or less than Π. Since the right heights are less than Π, we have mix-free forms of these derivations by the induction hypothesis. Let Π'_{21} and Π'_{22} be the mix-free derivations. Combining Π'_1, Π'_{21} and Π'_{22}, we get the following derivation, where Γ'_2 is $\Gamma_2 - X$.

$$\dfrac{\dfrac{\begin{matrix}\dfrac{\begin{matrix}\vdots\ \Pi'_1\\ \Gamma_1\,\Gamma'_2 \to C \supset D\,;\ A_1\,A_2\,\Delta_1 - X\end{matrix} \qquad \begin{matrix}\vdots\ \Pi'_{21}\\ A_1\,\Gamma'_2 \to C \supset D\,;\end{matrix}}{\Gamma_1\,\Gamma'_2\,(A_1\,\Gamma'_2 - A_1) \to C \supset D\,;\ (A_1\,A_2\,\Delta_1 - X) - A_1}(mix)\\ \vdots\ \text{structural rules}\end{matrix}}{\Gamma_1\,\Gamma'_2 \to C \supset D\,;\ A_2\,\Delta_1 - X} \qquad \begin{matrix}\vdots\ \Pi'_{22}\\ A_2\,\Gamma'_2 \to C \supset D\,;\end{matrix}}{\Gamma_1\,\Gamma'_2\,(A_2\,\Gamma'_2 - A_2) \to C \supset D\,;\ (A_2\,\Delta_1 - X) - A_2}(mix)$$

$$\dfrac{\vdots\ \text{structural rules}}{\Gamma_1\,\Gamma'_2 \to C \supset D\,;\ \Delta_1 - X}$$

Since the grades of the two (mix)-rules are both less than Π, we have a mix-free form of this derivation by the induction hypothesis.

Case 3: *The rule* $(r1)$ *is a right-logical rule whose principal formula is* X*, and* $(r2)$ *is neither* $(\rightarrow\supset)$*, nor a left-logical rule whose principal formula is* X*.* The proof is just similar to the one for LK or LJ.

Case 4: *The rule* $(r1)$ *is a right-logical rule whose principal formula is* X*, and* $(r2)$ *is a left-logical rule whose principal formula is also* X*.* The proof is also just similar to the cases of LJ and LK. $\square$

Unfortunately the computational behavior of the catch/throw mechanism is not captured by the cut-elimination process. Consider the following simple example.

$$\begin{array}{c}\dfrac{\dfrac{\dfrac{}{A \rightarrow A\,;}\,(init)}{A \rightarrow A\,;\,A}\,(throw)}{A \rightarrow A\,;}\,(catch)\\ <x,\ \mathbf{catch}\ u\ (\mathbf{throw}\ u\ x),\ >\end{array}$$

This is a cut-free derivation, but the realizer includes a catch-throw pair. Another kind of proof transformation should be considered to explain the computational behavior of the mechanism. Roughly, the throw operation corresponds to the following form of *non-local* proof transformation.

$$\dfrac{\begin{array}{c}\dfrac{\begin{array}{c}\vdots\ \Pi\\ \Gamma \rightarrow E\,;\,\Delta\end{array}}{\Gamma \rightarrow A\,;\,E\,\Delta}\,(throw)\\ \vdots\\ \hline \Gamma' \rightarrow E\,;\,E\,\Delta'\end{array}}{\Gamma' \rightarrow E\,;\,\Delta'}\,(catch) \quad\Longrightarrow\quad \begin{array}{c}\vdots\ \Pi\\ \Gamma \rightarrow E\,;\,\Delta\\ \hline \vdots\ (wxc \rightarrow wxc)\\ \hline \Gamma' \rightarrow E\,;\,\Delta'\end{array}\ ,$$

where $\Gamma' \rightarrow E\,;\,\Delta'$ must be derivable from $\Gamma \rightarrow E\,;\,\Delta$ by applying $(\rightarrow x)$, $(\rightarrow w)$, $(\rightarrow c)$, $(x\rightarrow)$, $(w\rightarrow)$ and $(c\rightarrow)$. We will discuss the details in Chapter 5.

Chapter 5

A natural extension with a non-determinism

5.1 A non-determinism by the catch and throw mechanism

In the previous chapters, the author showed that the catch/throw mechanism corresponds to a variant formulation of Gentzen's NJ following the Curry-Howard isomorphism in the opposite direction, and gave a correspondence with the conventional implementation by an abstract stack machine, in which the computational behavior of the mechanism was treated by a fixed evaluation strategy, the call-by-value strategy, and therefore the result of evaluation was unique. However, generally, the catch/throw mechanism introduces a non-determinism to evaluation processes, that is, the result of evaluation depends on the evaluation strategy. For example, let M be a term defined by

$$M = \mathbf{catch}\ u\ ((\lambda\, x.\, \lambda\, y.\, 1)\,(\mathbf{throw}\ u\ 2)\,(\mathbf{throw}\ u\ 3)).$$

There are three possible results for the evaluation of M depending on the evaluation strategy as follows.

$$\begin{aligned}
&M \to \mathbf{catch}\ u\ ((\lambda\, y.\, 1)\,(\mathbf{throw}\ u\ 3)) \to \mathbf{catch}\ u\ 1 \to 1 \\
&M \to \mathbf{catch}\ u\ (\mathbf{throw}\ u\ 2) \to 2 \\
&M \to \mathbf{catch}\ u\ (\mathbf{throw}\ u\ 3) \to 3
\end{aligned}$$

In this chapter, we first extend the language to capture this non-deterministic feature of the catch and throw mechanism, and introduce its typing system, and show that the new typing system has the subject reduction property.

5.2 A calculus with a non-determinism

We first extend the calculus described in Chapter 2 by new reduction rules. The syntax of the terms used in the new calculus is the same as the one defined in Chapter 2, but we do not use let-expressions such as **let** $x{=}M.\ N$. That is, *Term* is now redefined as:

$$\begin{array}{rl} \mathit{Term} ::= & \mathit{Const} \mid \mathit{Var} \\ \mid & \mathbf{throw}\ \mathit{Tvar}\ \mathit{Term} \mid \mathbf{catch}\ \mathit{Tvar}\ \mathit{Term} \\ \mid & \lambda\, \mathit{Var}.\mathit{Term}^* \mid \mathit{Term}\,\mathit{Term} \\ \mid & <\mathit{Term},\ \mathit{Term}> \mid \mathbf{proj_1}\,\mathit{Term} \mid \mathbf{proj_2}\,\mathit{Term} \\ \mid & \mathbf{inj_1}\,\mathit{Term} \mid \mathbf{inj_2}\,\mathit{Term} \mid \mathbf{case}\,\mathit{Term}\ \mathit{Var}.\mathit{Term}\ \mathit{Var}.\mathit{Term} \\ \mid & \kappa\ \mathit{Tvar}.\ \mathit{Term} \mid \mathit{Term}\,\mathit{Tvar}\ , \end{array}$$

We need not the notions of *values* or *evaluation contexts* in the new calculus.

5.2.1 Operational semantics

Now we define an operational semantics of the new calculus by a set of reduction rules on terms. The non-deterministic feature of the catch and throw mechanism is introduced by the following reduction rule.

Definition 5.2.1 ($\underset{\mathrm{t}}{\mapsto}$) A relation $\underset{\mathrm{t}}{\mapsto}$ on terms is defined as follows:

$$M[\mathbf{throw}\ u\ N/x] \underset{\mathrm{t}}{\mapsto} \mathbf{throw}\ u\ N \qquad (x \in FIV(M) \quad \text{and} \quad x \neq M)\,.$$

In other words,

$$\mathcal{C}[\mathbf{throw}\ u\ N] \underset{\mathrm{t}}{\mapsto} \mathbf{throw}\ u\ N,$$

where $\mathcal{C} \neq *$ and $\mathcal{C}$ does not capture any individual/tag variables occurring freely in **throw** u N. Note that N may not be a value.

Example 5.2.2

$$\begin{array}{rcl} <\mathbf{inj_1}\,(\mathbf{throw}\ u\ M),\ \mathbf{throw}\ v\ N> & \underset{\mathrm{t}}{\mapsto} & \mathbf{throw}\ u\ M \\ <\mathbf{inj_1}\,(\mathbf{throw}\ u\ M),\ \mathbf{throw}\ v\ N> & \underset{\mathrm{t}}{\mapsto} & \mathbf{throw}\ v\ N \\ \mathbf{throw}\ u\ M & \underset{\mathrm{t}}{\not\mapsto} & \mathbf{throw}\ u\ M \\ \mathbf{case}\ z\ x.(\mathbf{throw}\ u\ x)\ y.y & \underset{\mathrm{t}}{\not\mapsto} & \mathbf{throw}\ u\ x \\ \mathbf{catch}\ u\ (\mathbf{throw}\ u\ M) & \underset{\mathrm{t}}{\not\mapsto} & \mathbf{throw}\ u\ M \\ \mathbf{catch}\ v\ (\mathbf{throw}\ u\ (M\ v)) & \underset{\mathrm{t}}{\not\mapsto} & \mathbf{throw}\ u\ (M\ v) \end{array}$$

The rest of reduction rules is defined by the following rules.

Definition 5.2.3 ($\underset{\mathrm{n}}{\mapsto}$) A relation $\underset{\mathrm{n}}{\mapsto}$ on terms is defined as follows:

$$\begin{array}{rcll} \mathbf{catch}\ u\ M & \underset{\mathrm{n}}{\mapsto} & M & (u \notin FTV(M)) \\ \mathbf{catch}\ u\ (\mathbf{throw}\ u\ M) & \underset{\mathrm{n}}{\mapsto} & M & (u \notin FTV(M)) \\ (\lambda\, x.\, M)\, N & \underset{\mathrm{n}}{\mapsto} & M[N/x] & \\ (\kappa\, u.\, M)\, v & \underset{\mathrm{n}}{\mapsto} & M[v/u] & \\ \mathbf{proj_1}<M,\ N> & \underset{\mathrm{n}}{\mapsto} & M & \\ \mathbf{proj_2}<M,\ N> & \underset{\mathrm{n}}{\mapsto} & N & \\ \mathbf{case}\ (\mathbf{inj_1}\ L)\ x.M\ y.N & \underset{\mathrm{n}}{\mapsto} & M[L/x] & \\ \mathbf{case}\ (\mathbf{inj_2}\ L)\ x.M\ y.N & \underset{\mathrm{n}}{\mapsto} & N[L/y] & \end{array}$$

Definition 5.2.4 (Reduction rules) We define a relation $\mapsto$ by the union of $\underset{\mathrm{t}}{\mapsto}$ and $\underset{\mathrm{n}}{\mapsto}$, that is,

$$M \mapsto N \quad \text{iff} \quad M \underset{\mathrm{t}}{\mapsto} N \text{ or } M \underset{\mathrm{n}}{\mapsto} N.$$

Definition 5.2.5 ($\rightarrow$) We define a relation $\rightarrow$ as follows: $M \rightarrow N$ if and only if N is obtained from M by replacing an occurrence of M' in M by N' such that $M' \mapsto N'$. Let $\xrightarrow{*}$ be the transitive and reflexive closure of the relation $\rightarrow$.

Example 5.2.6 Let 1, 2 and 3 be distinct individual constants, and let M be as $M = \mathbf{catch}\ u\ ((\lambda x.\, \lambda y.\, 1)\, (\mathbf{throw}\ u\ 2)\, (\mathbf{throw}\ u\ 3))$.

$$\begin{aligned}
M &\rightarrow \mathbf{catch}\ u\ ((\lambda y.\, 1)\, (\mathbf{throw}\ u\ 3)) \rightarrow \mathbf{catch}\ u\ 1 \rightarrow 1 \\
M &\rightarrow \mathbf{catch}\ u\ (\mathbf{throw}\ u\ 2) \rightarrow 2 \\
M &\rightarrow \mathbf{catch}\ u\ (\mathbf{throw}\ u\ 3) \rightarrow 3
\end{aligned}$$

Definition 5.2.7 ($\xrightarrow{sub}$) We define a relation $\xrightarrow{sub}$ as follows:

$$M \xrightarrow{sub} N \quad \text{iff} \quad M \rightarrow N \text{ and } M \not\mapsto N,$$

and $\xrightarrow{sub*}$ as the transitive and reflexive closure of $\xrightarrow{sub}$.

5.3 Basic properties of the calculus

In this section, we consider about the basic properties of the calculus.

Proposition 5.3.1 (Extension of $\underset{\mathrm{CBV}}{\longrightarrow}$) *Let M and N be terms of the new calculus, that is, M or N does not include any let-expressions. If $M \underset{\mathrm{CBV}}{\xrightarrow{*}} N$, then $M \xrightarrow{*} N$.*

Proof. Obvious from the definitions of $\underset{\mathrm{CBV}}{\longrightarrow}$ and $\rightarrow$. □

Proposition 5.3.2 *Let L, M and N be terms and x an individual variable. Let u and v be tag variables. If $M \mapsto N$, then*

1. *$M[L/x] \mapsto N[L/x]$, and*
2. *$M[v/u] \mapsto N[v/u]$.*

Proof. Obvious from the definition of $\mapsto$. □

Proposition 5.3.3 *Let L, M and N be terms and x an individual variable. Let u and v be tag variables. If $M \rightarrow N$, then*

1. *$M[L/x] \rightarrow N[L/x]$, and*
2. *$M[v/u] \rightarrow N[v/u]$.*

Proof. We can assume that $M \xrightarrow{sub} N$ because it is obvious from Proposition 5.3.2 if $M \mapsto N$. We show that $M[L/x] \rightarrow N[L/x]$ by induction on $|M|$. Suppose $M \xrightarrow{sub} N$. Obviously, M must not be a variable. If $M = \mathbf{catch}\ u\ M'$ for some u and M', then we can assume that u is fresh, and that $N = \mathbf{catch}\ u\ N'$ for some N' such that $M' \rightarrow N'$. Since $M'[L/x] \xrightarrow{sub} N'[L/x]$ by the induction hypothesis, we get $M[L/x] \rightarrow N[L/x]$ in this case. We can similarly derive it even if M has one of the other forms. We can also get $M[v/u] \rightarrow N[v/u]$ by induction on $|M|$. □

Proposition 5.3.4 *Let M be a term, and let x and y be individual variables. Let u and v be tag variables. If $M[y/x, v/u] \mapsto N$, then $M \mapsto N'$ and $N = N'[y/x, v/u]$ for some N'.*

Proof. Obvious from the definition of $\mapsto$. □

Proposition 5.3.5 *If $M[y/x, v/u] \to N$, then $M \to N'$ and $N = N'[y/x, v/u]$ for some N'.*

Proof. Similar to the proof of Proposition 5.3.3. □

5.4 The typing system $L_{c/t}$

5.4.1 Syntax of typing judgements

We introduce a new typing system for the new calculus which has the extended reduction rules.

Definition 5.4.1 (Type expressions) We use the same class of type expressions, or formulas, as $L_{c/t}^{\text{CBV}}$. That is, atomic types, conjunctions $(A \wedge B)$, disjunctions $(A \vee B)$, implications $(A \supset B)$ and exceptions $(A \lhd B)$.

Definition 5.4.2 (Individual contexts) *Individual contexts* are also defined in the same way as $L_{c/t}^{\text{CBV}}$.

Definition 5.4.3 (Tag contexts) We extend the definition of tag contexts as follows. A *tag context* is a finite mapping which assigns a pair of a type expression and a set of individual variables to each tag variable in its domain. We use $\Delta, \Delta', \ldots$ to denote tag contexts. Let $u_1, \ldots, u_n$ be tag variables. Let $B_1, \ldots, B_n$ be type expressions, and let $V_1, \ldots, V_n$ be sets of individual variables such that if $u_i = u_j$ then $B_i = B_j$ and $V_i = V_j$ for any i and j. We use $\{u_1 : B_1^{V_1}, \ldots, u_n : B_n^{V_n}\}$ to denote a tag context whose domain is $\{u_1, \ldots, u_n\}$ and which assigns the pair $< B_i, V_i >$ to u_i for each i. We denote the first and the second components of $\Delta(u)$ by $\Delta^t(u)$ and $\Delta^v(u)$, respectively. For example, $\Delta^t(u_i) = B_i$ and $\Delta^v(u_i) = V_i$ if $\Delta = \{u_1 : B_1^{V_1}, \ldots, u_n : B_n^{V_n}\}$.

Definition 5.4.4 (Compatible contexts) Let Γ and Γ' be individual contexts. Γ is *compatible* with Γ' if and only if $\Gamma(x) = \Gamma'(x)$ for any individual variable $x \in Dom(\Gamma) \cap Dom(\Gamma')$. We denote it by $\Gamma \parallel \Gamma'$. Note that $\Gamma \cup \Gamma'$ is also an individual context if $\Gamma \parallel \Gamma'$. The compatibility of tag contexts is also defined as follows: Δ is *compatible* with Δ' if and only if $\Delta^t(u) = \Delta'^t(u)$ for any individual variable $u \in Dom(\Delta) \cap Dom(\Delta')$. We denote it by $\Delta \parallel \Delta'$. When Δ and Δ' are compatible, we define a new tag context $\Delta \sqcup \Delta'$ as follows.

$$(\Delta \sqcup \Delta')(u) = \begin{cases} (\Delta^t(u),\ \Delta^v(u) \cup \Delta'^v(u)) & \text{if } u \in Dom(\Delta) \cap Dom(\Delta') \\ \Delta(u) & \text{if } u \in Dom(\Delta) \text{ and } u \notin Dom(\Delta') \\ \Delta'(u) & \text{if } u \notin Dom(\Delta) \text{ and } u \in Dom(\Delta') \end{cases}$$

Note that $Dom(\Delta \sqcup \Delta') = Dom(\Delta) \cup Dom(\Delta')$.

Definition 5.4.5 Let Δ be as $\Delta = \{u_1 : B_1^{V_1}, \ldots, u_n : B_n^{V_n}\}$, and let u and v be tag variables. If $\{u, v\} \subset Dom(\Delta)$ implies $\Delta^t(u) = \Delta^t(v)$, then we define a tag context $\Delta[v/u]$ as follows.

$$\Delta[v/u] = \{u_1[v/u] : B_1^{V_1}, \ldots, u_n[v/u] : B_n^{V_n}\}.$$

We define $\Gamma[y/x]$ similarly for an individual context Γ and individual variables x and y.

Definition 5.4.6 Let V be a set of individual variables. We define a tag context $\Delta[V/\{x\}]$ as follows.

$$\begin{aligned}
&Dom(\Delta[V/\{x\}]) = Dom(\Delta)\\
&\Delta[V/\{x\}]^t(u) = \Delta^t(u)\\
&\Delta[V/\{x\}]^v(u) = \begin{cases} (\Delta^v(u) - \{x\}) \cup V & \text{if } x \in \Delta^v(u) \\ \Delta^v(u) & \text{otherwise.} \end{cases}
\end{aligned}$$

Definition 5.4.7 (Typing judgements) Let Γ and Δ be an individual context and a tag context, respectively, such that $\Delta^v(u) \subset Dom(\Gamma)$ for any $u \in Dom(\Delta)$. Let M be a term, and C a type expression. *Typing judgements* have the following form.

$$\Gamma \vdash M : C \,;\, \Delta$$

The intended meaning of a typing judgement $\{x_1 : A_1, \ldots, x_m : A_m\} \vdash M : C;\ \{u_1 : B_1^{V_1}, \ldots, u_n : B_n^{V_n}\}$ is roughly that when we execute the program M supplying values of the types $A_1 \ldots A_m$ for the corresponding free variables $x_1, \ldots, x_m$ of M, it normally reduces to a value of the type C, otherwise the program throws a value of B_j with a tag u_j for some j $(1 \leq j \leq n)$, and the thrown value depends on only the individual variables which belong to V_j.

5.4.2 $L_{c/t}$

We denote the typing system by $L_{c/t}$, which can be an extension of $L_{c/t}^{\text{CBV}}$.

Definition 5.4.8 (Typing rules) $L_{c/t}$ is defined by the following set of typing rules.

$$\frac{}{\Gamma \cup \{x : A\} \vdash x : A \,;\, \Delta}\ (var)$$

$$\frac{\Gamma \vdash M : A \,;\, \Delta \sqcup \{u : A^V\}}{\Gamma \vdash \mathbf{catch}\ u\ M : A \,;\, \Delta}\ (catch)$$

$$\frac{\Gamma_1 \vdash M : E \,;\, \Delta}{\Gamma_1 \cup \Gamma_2 \vdash \mathbf{throw}\ u\ M : A \,;\, \Delta \sqcup \{u : E^{Dom(\Gamma_1)}\}}\ (throw)$$

$$\frac{\Gamma \cup \{x : A\} \vdash M : B \,;\, \Delta}{\Gamma \vdash \lambda x.\, M : A \supset B \,;\, \Delta}\ (\supset\text{-I})\quad (x \notin \Delta^v(u) \text{ for any } u \in Dom(\Delta))$$

$$\frac{\Gamma_1 \vdash M : A \supset B \,;\, \Delta_1 \quad \Gamma_2 \vdash N : A \,;\, \Delta_2}{\Gamma_1 \cup \Gamma_2 \vdash M\,N : B \,;\, \Delta_1 \sqcup \Delta_2}\ (\supset\text{-E})$$

$$\frac{\Gamma \vdash M : A \,;\, \Delta \sqcup \{u : E^V\}}{\Gamma \vdash \kappa u.\, M : A \triangleleft E \,;\, \Delta}\ (\triangleleft\text{-I})$$

$$\frac{\Gamma_1 \vdash M : A \triangleleft E \,;\, \Delta}{\Gamma_1 \cup \Gamma_2 \vdash M\, u : A \,;\, \Delta \sqcup \{u : E^{Dom(\Gamma_1)}\}}\ (\triangleleft\text{-E})$$

$$\frac{\Gamma_1 \vdash M : A \,;\, \Delta_1 \quad \Gamma_2 \vdash N : B \,;\, \Delta_2}{\Gamma_1 \cup \Gamma_2 \vdash \langle M,\ N\rangle : A \wedge B \,;\, \Delta_1 \sqcup \Delta_2}\ (\wedge\text{-I})$$

$$\frac{\Gamma \vdash M : A \wedge B \,;\, \Delta}{\Gamma \vdash \mathbf{proj_1}\, M : A \,;\, \Delta}\ (\wedge_1\text{-E}) \qquad \frac{\Gamma \vdash M : A \wedge B \,;\, \Delta}{\Gamma \vdash \mathbf{proj_2}\, M : B \,;\, \Delta}\ (\wedge_2\text{-E})$$

$$\frac{\Gamma \vdash M : A \,;\, \Delta}{\Gamma \vdash \mathbf{inj_1}\, M : A \vee B \,;\, \Delta}\ (\vee_1\text{-I}) \qquad \frac{\Gamma \vdash M : B \,;\, \Delta}{\Gamma \vdash \mathbf{inj_2}\, M : A \vee B \,;\, \Delta}\ (\vee_2\text{-I})$$

$$\frac{\Gamma_1 \vdash L : A \vee B \,;\, \Delta_1 \quad \Gamma_2 \cup \{x : A\} \vdash M : C \,;\, \Delta_2 \quad \Gamma_3 \cup \{y : B\} \vdash N : C \,;\, \Delta_3}{\begin{array}{l}\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \vdash \mathbf{case}\ L\ x.M\ y.N : C \,; \\ \qquad \Delta_1 \sqcup \Delta_2[Dom(\Gamma_1)/\{x\}] \sqcup \Delta_3[Dom(\Gamma_1)/\{y\}]\end{array}}\ (\vee\text{-E})$$

The side condition for ($\supset$-I) is necessary to keep the system constructive. Note that the following inference rule of $L^{\mathrm{CBV}}_{c/t}$ corresponds to ($\supset$-I) of $L_{c/t}$.

$$\frac{\Gamma \cup \{x : A\} \vdash M : B \,;\, \{\}}{\Gamma \vdash \lambda\, x.\, M : A \supset B \,;\, \{\}}\ (\supset\text{-I})$$

As a logic, ($\supset$-I) of $L^{\mathrm{CBV}}_{c/t}$ is equivalent to ($\supset$-I) of Definition 5.4.8, but is too restrictive with respect to the variation of proofs, that is, typed programs. For example, the following typing judgement, which is derivable in $L_{c/t}$, would not be derivable if we replaced ($\supset$-I) by the one of $L^{\mathrm{CBV}}_{c/t}$.

$$\{\} \vdash \mathbf{catch}\ u\ (\lambda\, x.\, \mathbf{throw}\ u\ (\lambda\, y.\, y)) : A \supset A \,;\, \{\}$$

Moreover, the language would not have a subject reduction property, because

$$\{\} \vdash \mathbf{catch}\ u\ ((\lambda\, z.\, \lambda\, x.\, z)\, (\mathbf{throw}\ u\ (\lambda\, y.\, y))) : A \supset A \,;\, \{\}$$

would be still derivable, but

$$\mathbf{catch}\ u\ ((\lambda\, z.\, \lambda\, x.\, z)\, (\mathbf{throw}\ u\ (\lambda\, y.\, y))) \to \mathbf{catch}\ u\ (\lambda\, x.\, \mathbf{throw}\ u\ (\lambda\, y.\, y)).$$

This is the reason why we maintain the set of the relevant individual variables to each tag in tag contexts of typing judgements.

The following example of a derivation shows that the calculus does not have Church-Rosser property even if we consider only the well-typed terms. Let M be the term $\lambda\, x.\, \lambda\, f.\, \mathbf{catch}\ u\ ((\lambda\, y.\, x)\, (\mathbf{throw}\ u\ (f\ x)))$. The well-typed term M has two normal forms as follows.

$$M \to \lambda\, x.\, \lambda\, f.\, \mathbf{catch}\ u\ (\mathbf{throw}\ u\ (f\ x)) \to \lambda\, x.\, \lambda\, f.\, f\ x$$
$$M \to \lambda\, x.\, \lambda\, f.\, \mathbf{catch}\ u\ x \to \lambda\, x.\, \lambda\, f.\, x$$

Example 5.4.9 Let Γ be as $\Gamma = \{x:A, f:A\supset A\}$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\{y:B\}\vdash x:A\,;\,\{\}}\,(var)}{\{\}\vdash \lambda\,y.\,x:B\supset A\,;\,\{\}}\,(\supset\text{-I})
\quad
\dfrac{\dfrac{\dfrac{}{\Gamma\vdash f:A\supset A\,;\,\{\}}\,(var)\quad \dfrac{}{\Gamma\vdash x:A\,;\,\{\}}\,(var)}{\Gamma\vdash f\,x:A\,;\,\{\}}\,(\supset\text{-E})}{\Gamma\vdash \mathbf{throw}\;u\,(f\,x):B\,;\,\{u:A^{\{x,f\}}\}}\,(throw)}
{\dfrac{\Gamma\vdash(\lambda\,y.\,x)\,(\mathbf{throw}\;u\,(f\,x)):A\,;\,\{u:A^{\{x,f\}}\}}{\Gamma\vdash \mathbf{catch}\;u\,((\lambda\,y.\,x)\,(\mathbf{throw}\;u\,(f\,x))):A\,;\,\{\}}\,(catch)}\,(\supset\text{-E})}
{\dfrac{\{x:A\}\vdash\lambda\,f.\,\mathbf{catch}\;u\,((\lambda\,y.\,x)\,(\mathbf{throw}\;u\,(f\,x))):(A\supset A)\supset A\,;\,\{\}}{\{\}\vdash\lambda\,x.\,\lambda\,f.\,\mathbf{catch}\;u\,((\lambda\,y.\,x)\,(\mathbf{throw}\;u\,(f\,x))):A\supset(A\supset A)\supset A\,;\,\{\}}\,(\supset\text{-I})}\,(\supset\text{-I})
$$

5.4.3 Basic properties of $L_{c/t}$

In this subsection, we presents a some basic properties of the system as a preparation for proving the subject reduction property of $L_{c/t}$.

Proposition 5.4.10 *If $\Gamma\vdash M:C\,;\,\Delta$ is derivable, then $FIV(M)\subset Dom(\Gamma)$ and $FTV(M)\subset Dom(\Delta)$.*

Proof. By induction on the derivation of $\Gamma\vdash M:C\,;\,\Delta$. □

Definition 5.4.11 Let Δ and Δ' be tag contexts. We define a relation $\Delta\sqsubset\Delta'$ as follows. The relation $\Delta\sqsubset\Delta'$ holds if and only if

- $Dom(\Delta)\subset Dom(\Delta')$, and
- $\Delta^t(u)=\Delta'^t(u)$ and $\Delta^v(u)\subset\Delta'^v(u)$ for any $u\in Dom(\Delta)$.

Note that $\Delta\sqsubset(\Delta\sqcup\Delta')$ if $\Delta\parallel\Delta'$.

Definition 5.4.12 Let d be a natural number. We say a typing judgement is *d-derivable* if there exists a derivation of the judgement whose depth is less than or equal to d.

Proposition 5.4.13 *Let d be a natural number, and let $\Gamma\vdash M:C\,;\,\Delta$ be a d-derivable typing judgement.*

1. *If $\Gamma\subset\Gamma'$ and $\Delta\sqsubset\Delta'$, then $\Gamma'\vdash M:C\,;\,\Delta'$ is also d-derivable.*
2. *If $\Gamma[y/x]$ is well defined, then $\Gamma[y/x]\vdash M[y/x]:C\,;\,\Delta[\{y\}/\{x\}]$ is also d-derivable.*
3. *If $\Delta[v/u]$ is well defined, then $\Gamma\vdash M[v/u]:C\,;\,\Delta[v/u]$ is also d-derivable.*

Proof. By simultaneous inductions on d. □

Proposition 5.4.14 *Let x and u be as $x\notin FIV(M)$ and $u\notin FTV(M)$.*

1. *If $\Gamma\cup\{x:A\}\vdash M:C\,;\,\Delta$ is derivable, then $\Gamma\vdash M:C\,;\,\Delta[\{\}/\{x\}]$ is also derivable.*
2. *If $\Gamma\vdash M:C\,;\,\Delta\sqcup\{u:E^V\}$ is derivable, then $\Gamma\vdash M:C\,;\,\Delta$ is also derivable.*

Proof. Straightforward induction on the derivations. □

Proposition 5.4.15 (Extension of $L^{\text{CBV}}_{c/t}$) *Let M be a term of the new calculus, that is, M does not include any let-expressions. Let Γ be a tag context, and C a formula. Let $u_1, \ldots, u_n$ be a sequence of tag variables, and let $B_1, \ldots, B_n$ be a sequence of formulas. If $\Gamma \vdash M : C\,;\, \{u_1 : B_1, \ldots, u_n : B_n\}$ is derivable in $L^{\text{CBV}}_{c/t}$, then $\Gamma \vdash M : C\,;\, \{u_1 : B_1^{Dom(\Gamma)}, \ldots, u_n : B_n^{Dom(\Gamma)}\}$ is derivable in $L_{c/t}$.*

Proof. By induction on the depth of the derivation of $\Gamma \vdash M : C\,;\, \{u_1 : B_1, \ldots, u_n : B_n\}$. Use Proposition 5.4.13. □

Proposition 5.4.16 (Throw) *Let M be term, and let u be a tag variable. If $\Gamma \vdash$ **throw** u $M : C\,;\, \Delta$ is derivable, then $\Gamma \vdash$ **throw** u $M : A\,;\, \Delta$ is also derivable for any type A.*

Proof. Since $\Gamma \vdash$ **throw** u $M : C\,;\, \Delta$ is derivable, so is $\Gamma \vdash M : E\,;\, \Delta'$ for some E and Δ' such that $\Delta = \Delta' \sqcup \{u : E^{Dom(\Gamma)}\}$. Therefore, we can derive $\Gamma \vdash$ **throw** u $M : A\,;\, \Delta$ for any A by $(throw)$. □

Proposition 5.4.17 (Substitution) *Let Γ_1, Γ_2, Δ_1 and Δ_2 be as $\Gamma_1 \parallel \Gamma_2$ and $\Delta_1 \parallel \Delta_2$. If $\Gamma_1 \vdash N : A\,;\, \Delta_1$ and $\Gamma_2 \cup \{x : A\} \vdash M : C\,;\, \Delta_2$ are derivable, then $\Gamma_1 \cup \Gamma_2 \vdash M[N/x] : C\,;\, \Delta_1 \sqcup \Delta_2[Dom(\Gamma_1)/\{x\}]$ is also derivable.*

Proof. By induction on the depth of the derivation of $\Gamma_2 \cup \{x : A\} \vdash M : C\,;\, \Delta_2$. Suppose that $\Gamma_1 \vdash N : A\,;\, \Delta_1$ and $\Gamma_2 \cup \{x : A\} \vdash M : C\,;\, \Delta_2$ are derivable. First, suppose also that $x \notin FIV(M)$, that is, $M[N/x] = M$. Since $\Gamma_2 \cup \{x : A\} \vdash M : C\,;\, \Delta_2$ is derivable, so is $\Gamma_2 \vdash M : C\,;\, \Delta_2[\{\}/\{x\}]$ by Proposition 5.4.14, and this implies that $\Gamma_1 \cup \Gamma_2 \vdash M : C\,;\, \Delta_1 \sqcup \Delta_2[Dom(\Gamma_1)/\{x\}]$ is derivable by Proposition 5.4.13. Therefore, we now assume that $x \in FIV(M)$. By cases on the last rule used in the derivation of $\Gamma_2 \cup \{x : A\} \vdash M : C\,;\, \Delta_2$.

Case 1: *The last rule is (var).* That is, $M = x$ since $x \in FIV(M)$. We can derive $\Gamma_1 \cup \Gamma_2 \vdash M[N/x] : C\,;\, \Delta_1 \sqcup \Delta_2[Dom(\Gamma_1)/\{x\}]$ by applying Proposition 5.4.13 to the derivation of $\Gamma_1 \vdash N : A\,;\, \Delta_1$ since $M[N/x] = N$ and $C = A$.

Case 2: *The last rule is $(catch)$.* In this case, $M =$ **catch** u M' and the following judgement is derivable for some u, V and M'.

$$\Gamma_2 \cup \{x : A\} \vdash M' : C\,;\, \Delta_2 \sqcup \{u : C^V\}$$

We can assume that $u \notin Dom(\Delta_1)$ by Proposition 5.4.13. By the induction hypothesis, we have a derivation of

$$\Gamma_1 \cup \Gamma_2 \vdash M'[N/x] : C\,;\, \Delta_1 \sqcup (\Delta_2 \sqcup \{u : C^V\})[Dom(\Gamma_1)/\{x\}]. \tag{5.1}$$

Since $u \notin Dom(\Delta_1)$, we get $M[N/x] =$ **catch** u $(M'[N/x])$. By applying $(catch)$ to (5.1), we get $\Gamma_1 \cup \Gamma_2 \vdash M[N/x] : C\,;\, \Delta_1 \sqcup \Delta_2[Dom(\Gamma_1)/\{x\}]$.

Case 3: *The last rule is $(throw)$.* In this case, $M =$ **throw** u M' and the following judgement is derivable for some u, M', E, Γ_2' and Δ such that $\Gamma_2' \subset \Gamma_2 \cup \{x : A\}$ and $\Delta_2 = \Delta \sqcup \{u : E^{Dom(\Gamma_2')}\}$.

$$\Gamma_2' \vdash M' : E\,;\, \Delta$$

We get $x \in FIV(M')$, that is, $x \in Dom(\Gamma_2')$ from $x \in FIV(M)$. Let Γ be as $\Gamma = \Gamma_2' - \{x : A\}$, that is, $\Gamma_2' = \Gamma \cup \{x : A\}$. By the induction hypothesis, we have a derivation of

$$\Gamma_1 \cup \Gamma \vdash M'[N/x] : E\,;\ \Delta_1 \sqcup \Delta[Dom(\Gamma_1)/\{x\}].$$

Since $M[N/x] = \mathbf{throw}\ u\ (M'[N/x])$, by applying $(throw)$,

$$\Gamma_1 \cup \Gamma \vdash M[N/x] : C\,;\ \Delta_1 \sqcup \Delta[Dom(\Gamma_1)/\{x\}] \sqcup \{u : E^{Dom(\Gamma_1 \cup \Gamma)}\}.$$

Since $\Gamma \subset \Gamma_2$, by Proposition 5.4.13 again,

$$\Gamma_1 \cup \Gamma_2 \vdash M[N/x] : C\,;\ \Delta_1 \sqcup \Delta[Dom(\Gamma_1)/\{x\}] \sqcup \{u : E^{Dom(\Gamma_1 \cup \Gamma)}\}.$$

Note that

$$\Delta[Dom(\Gamma_1)/\{x\}] \sqcup \{u : E^{Dom(\Gamma_1 \cup \Gamma)}\} = \Delta_2[Dom(\Gamma_1)/\{x\}]$$

because $\Delta_2 = \Delta \sqcup \{u : E^{Dom(\Gamma) \cup \{x\}}\}$ and $x \notin Dom(\Gamma)$.

Case 4: *The last rule is* $(\supset\text{-}I)$. In this case $M = \lambda y.\, M'$, $C = C_1 \supset C_2$ and the following judgement is derivable for some y, C_1, C_2 and M' such that $y \notin \Delta_2^v(u)$ for any $u \in Dom(\Delta_2)$.

$$\Gamma_2 \cup \{x : A\} \cup \{y : C_1\} \vdash M' : C_2\,;\ \Delta_2$$

We can assume that $y \notin Dom(\Gamma_1)$ by Proposition 5.4.13, and get $M'[N/x] = \lambda y.\, (M[N/x])$. By the induction hypothesis, we have a derivation of

$$\Gamma_1 \cup \Gamma_2 \cup \{y : C_1\} \vdash M'[N/x] : C_2\,;\ \Delta_1 \sqcup \Delta_2[Dom(\Gamma_1)/\{x\}]. \tag{5.2}$$

We get $y \notin (\Delta_1 \sqcup \Delta_2[Dom(\Gamma_1)/\{x\}])^v(u)$ for any $u \in Dom(\Delta_1 \sqcup \Delta_2[Dom(\Gamma_1)/\{x\}])$ since $y \notin \Delta_2^v(u)$ for any $u \in Dom(\Delta_2)$ and $y \notin Dom(\Gamma_1)$. Therefore, we can derive

$$\Gamma_1 \cup \Gamma_2 \vdash \lambda y.\, (M'[N/x]) : C_2\,;\ \Delta_1 \sqcup \Delta_2[Dom(\Gamma_1)/\{x\}]$$

by applying ($\supset$-I) to (5.2).

Case 5: *The last rule is one of the rest.* Similar. □

5.5 The subject reduction property of $L_{c/t}$

As mentioned in Section 5.4.2, the calculus does not have Church-Rosser property even if we consider only the well-typed terms. However, it has the subject reduction property, which compensates for this unpleasant feature. In this section, we show the subject reduction property of $L_{c/t}$.

Lemma 5.5.1 *If* $\Gamma \vdash M : C\,;\ \Delta$ *is derivable and* $M \underset{t}{\mapsto} \mathbf{throw}\ v\ N$, *then* $\Gamma \vdash \mathbf{throw}\ v\ N : C\,;\ \Delta$ *is also derivable.*

Proof. By induction on the depth of the derivation of $\Gamma \vdash M : C\,;\ \Delta$. Suppose that $\Gamma \vdash M : C\,;\ \Delta$ is derivable and $M \underset{t}{\mapsto} \mathbf{throw}\ v\ N$. By Proposition 5.4.16, it is enough to show that $\Gamma \vdash \mathbf{throw}\ v\ N : C'\,;\ \Delta$ is derivable for some C'. By cases according to the last rules used in the derivation.

Case 1: *The last rule is* (var). This is impossible because $M \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$.

Case 2: *The last rule is* $(catch)$. $M = \textbf{catch}\ u\ M'$ and the following judgement is derivable for some u, V and M'.

$$\Gamma \vdash M' : C\,;\ \Delta \sqcup \{u : C^V\} \tag{5.3}$$

Since $M \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$, we get $u \notin FTV(\textbf{throw}\ v\ N)$ and

$$M' = \textbf{throw}\ v\ N \quad \text{or} \quad M' \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N.$$

Therefore, from (5.3) or the induction hypothesis on (5.3),

$$\Gamma \vdash \textbf{throw}\ v\ N : C\,;\ \Delta \sqcup \{u : C^V\}.$$

We get $\Gamma \vdash \textbf{throw}\ v\ N : C\,;\ \Delta$ by Proposition 5.4.14 since $u \notin FTV(\textbf{throw}\ v\ N)$.

Case 3: *The last rule is* $(throw)$. In this case, $M = \textbf{throw}\ u\ M'$ and the following judgement is derivable for some u, M', E, Γ' and Δ' such that $\Gamma' \subset \Gamma$ and $\Delta = \Delta' \sqcup \{u : E^{Dom(\Gamma')}\}$.

$$\Gamma' \vdash M' : E\,;\ \Delta' \tag{5.4}$$

We get $M' = \textbf{throw}\ v\ N$ or $M' \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$ from $M \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$. Therefore, from (5.4) or the induction hypothesis on (5.4),

$$\Gamma' \vdash \textbf{throw}\ v\ N : E\,;\ \Delta'.$$

We get $\Gamma \vdash \textbf{throw}\ v\ N : E\,;\ \Delta$ by Proposition 5.4.13 since $\Gamma' \subset \Gamma$ and $\Delta' \sqsubset \Delta$.

Case 4: *The last rule is* $(\supset\text{-}I)$. $M = \lambda x.\, M'$, $C = C_1 \supset C_2$ and the following judgement is derivable for some x, C_1, C_2 and M' such that $x \notin \Delta^v(u)$ for any $u \in Dom(\Delta)$.

$$\Gamma \cup \{x : C_1\} \vdash M' : C_2\,;\ \Delta \tag{5.5}$$

Since $M \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$, we get $x \notin FIV(\textbf{throw}\ v\ N)$ and

$$M' = \textbf{throw}\ v\ N \quad \text{or} \quad M' \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N.$$

Therefore, from (5.5) or the induction hypothesis on (5.5),

$$\Gamma \cup \{x : C_1\} \vdash \textbf{throw}\ v\ N : C_2\,;\ \Delta.$$

We get $\Gamma \vdash \textbf{throw}\ v\ N : C_2\,;\ \Delta$ by Proposition 5.4.14 since $x \notin FIV(\textbf{throw}\ v\ N)$.

Case 5: *The last rule is one of the rest.* Similar to Case 2 and Case 3. $\square$

Lemma 5.5.2 *If* $\Gamma \vdash M : C\,;\ \Delta$ *is derivable and* $M \underset{\mathrm{n}}{\mapsto} N$, *then* $\Gamma \vdash N : C\,;\ \Delta$ *is also derivable.*

Proof. By induction on the depth of the derivation of $\Gamma \vdash M : C\,;\ \Delta$. Suppose that $\Gamma \vdash M : C\,;\ \Delta$ is derivable and $M \underset{\mathrm{n}}{\mapsto} N$. By cases according to the form of M.

Case 1: $M = \textbf{catch}\ u\ N$ *and* $u \notin FTV(N)$. In this case, $\Gamma \vdash N : C\,;\ \Delta \sqcup \{u : C^V\}$ is derivable for some V. We get $\Gamma \vdash N : C\,;\ \Delta$ by Proposition 5.4.14 since $u \notin FTV(N)$.

Case 2: *$M =$ **catch** u (**throw** u N) and $u \notin FTV(N)$.* The following judgement is derivable for some V, Γ' and Δ' such that $\Gamma' \subset \Gamma$ and $\Delta \sqcup \{u : C^V\} = \Delta' \sqcup \{u : C^{Dom(\Gamma')}\}$.

$$\Gamma' \vdash N : C\,;\ \Delta'$$

Since $\Gamma' \subset \Gamma$ and $\Delta' \sqsubset \Delta \sqcup \{u : C^V\}$, $\Gamma \vdash N : C\,;\ \Delta \sqcup \{u : C^V\}$ is derivable by Proposition 5.4.13. Therefore, $\Gamma \vdash N : C\,;\ \Delta$ is also derivable by Proposition 5.4.14 since $u \notin FTV(N)$.

Case 3: *$M = (\lambda x.\, M_1)\, M_2$ and $N = M_1[M_2/x]$ for some x, M_1 and M_2.* The following two judgements are derivable for some A, Γ_1, Γ_2, Δ_1 and Δ_2 such that $\Gamma = \Gamma_1 \cup \Gamma_2$, $\Delta = \Delta_1 \sqcup \Delta_2$ and $x \notin \Delta_1^v(u)$ for any $u \in Dom(\Delta_1)$.

$$\Gamma_1 \cup \{x : A\} \vdash M_1 : C\,;\ \Delta_1 \tag{5.6}$$

$$\Gamma_2 \vdash M_2 : A\,;\ \Delta_2 \tag{5.7}$$

Therefore, by Lemma 5.4.17, we get $\Gamma \vdash M_1[M_2/x] : C\,;\ \Delta_1[Dom(\Gamma_2)/\{x\}] \sqcup \Delta_2$ from (5.6) and (5.7), where $\Delta_1[Dom(\Gamma_2)/\{x\}] \sqcup \Delta_2 = \Delta_1 \sqcup \Delta_2 = \Delta$ since $x \notin \Delta_1^v(u)$ for any $u \in Dom(\Delta_1)$.

Case 4: *$M = (\kappa\, u.\, M')\, v$ and $N = M'[v/u]$ for some u, v and M'.* The following judgement is derivable for some E, Γ', Δ' and V such that $\Gamma' \subset \Gamma$ and $\Delta = \Delta' \sqcup \{v : E^{Dom(\Gamma')}\}$.

$$\Gamma' \vdash M' : C\,;\ \Delta' \sqcup \{u : E^V\}$$

Since $\Delta' \parallel \{v : E^{Dom(\Gamma')}\}$, $\Gamma' \vdash M'[v/u] : C\,;\ \Delta'[v/u] \cup \{v : E^V\}$ is derivable by Proposition 5.4.13. Since $\Gamma' \subset \Gamma$, by Proposition 5.4.13 again,

$$\Gamma \vdash M'[v/u] : C\,;\ \Delta'[v/u] \cup \{v : E^V\}.$$

Since $V \subset Dom(\Gamma')$ and $\Delta'^v(u) \subset Dom(\Gamma')$,

$$\Delta'[v/u] \sqcup \{v : E^V\} \sqsubset \Delta'[v/u] \sqcup \{v : E^{Dom(\Gamma')}\} \sqsubset \Delta' \sqcup \{v : E^{Dom(\Gamma')}\} = \Delta.$$

Therefore, $\Gamma \vdash M'[v/u] : C\,;\ \Delta$ is derivable by Proposition 5.4.13.

Case 5: *$M =$ **proj**$_i$ $<M_1,\ M_2>$ and $N = M_i$ for some i ($i = 1, 2$).* Similar.

Case 6: *$M =$ **case** (**inj**$_i$ M_0) $x_1.M_1$ $x_2.M_2$ and $N = M_i[M_0/x_i]$ for some i ($i = 1, 2$).* Similar. □

Lemma 5.5.3 *If $\Gamma \vdash M : C\,;\ \Delta$ is derivable and $M \mapsto N$, then $\Gamma \vdash N : C\,;\ \Delta$ is also derivable.*

Proof. Straightforward from Lemma 5.5.1 and Lemma 5.5.2. □

Theorem 5.5.4 (Subject reduction) *If $\Gamma \vdash M : C\,;\ \Delta$ is derivable and $M \to N$, then $\Gamma \vdash N : C\,;\ \Delta$ is also derivable.*

Proof. By induction on the depth of the derivation of $\Gamma \vdash M : C\,;\ \Delta$. Suppose that $\Gamma \vdash M : C\,;\ \Delta$ is derivable and $M \to N$. If $M \mapsto N$, then trivial by Lemma 5.5.3. Therefore we can assume that $M \rightarrow N$ and $M \not\mapsto N$. By cases according to the last rules used in the derivation. A typical one is the case that the last rule is $(throw)$. In this case, $M = \textbf{throw}\ u\ M'$ and

$$\Gamma' \vdash M' : E\ ;\ \Delta'$$

is derivable for some u, M', E, Γ' and Δ' such that $\Gamma' \subset \Gamma$ and $\Delta = \Delta' \sqcup \{u : E^{Dom(\Gamma')}\}$. Since $M \rightarrow N$ and $M \not\mapsto N$, $M' \rightarrow N'$ and $N = \textbf{throw}\ u\ N'$ for some N'. Therefore, $\Gamma' \vdash N' : E\ ;\ \Delta'$ is derivable by the induction hypothesis. We get $\Gamma \vdash \textbf{throw}\ u\ N' : E\ ;\ \Delta$ by applying $(throw)$. The proofs for other cases are just similar. $\square$

5.6 $L_{c/t}$ as a logic

In this section, we show that $L_{c/t}$ can be regarded as a conservative extension of the propositional fragment of the standard intuitionistic logic such as Gentzen's NJ and LJ. We first translate each typing judgement of $L_{c/t}$ into a formula, and then show its provability in the standard intuitionistic logic. Let $\leq$ be a total ordering over the union of Var and $Tvar$ through this section.

Definition 5.6.1 Let Δ be a tag context, and u a tag variable. We define a tag context $\Delta \backslash u$ as follows.

$$\begin{array}{rcll} Dom(\Delta\backslash u) & = & Dom(\Delta) - \{u\} & \\ (\Delta\backslash u)(v) & = & \Delta(v) & (v \in Dom(\Delta) - \{u\}) \end{array}$$

Let Γ be an individual context, and x an individual variable. $\Gamma\backslash x$ is defined in the same way.

Definition 5.6.2 Let Δ be a tag context, and x an individual variable. We define two tag contexts $Dep(x, \Delta)$ and $Indep(x, \Delta)$ as follows.

$$\begin{array}{rcl} Dep(x, \{\}) & = & Indep(x, \{\}) \ = \ \{\} \\ Dep(x, \{u : A^V\} \cup \Delta) & = & \left\{ \begin{array}{ll} \{u : A^V\} \cup Dep(x, \Delta\backslash u) & \text{if } x \in V \\ Dep(x, \Delta\backslash u) & \text{otherwise} \end{array} \right. \\ Indep(x, \{u : A^V\} \cup \Delta) & = & \left\{ \begin{array}{ll} \{u : A^V\} \cup Indep(x, \Delta\backslash u) & \text{if } x \notin V \\ Indep(x, \Delta\backslash u) & \text{otherwise} \end{array} \right. \end{array}$$

Definition 5.6.3 Let Δ be a non-empty tag context. We define a formula $Disj(\Delta)$ as follows. Let u be the tag variable such that $u \in Dom(\Delta)$ and $u \leq v$ for any $v \in Dom(\Delta)$. Then,

$$Disj(\Delta) = \left\{ \begin{array}{ll} \Delta^t(u) & \text{if } Dom(u) \text{ is a singleton} \\ \Delta^t(u) \vee Disj(\Delta\backslash u) & \text{otherwise.} \end{array} \right.$$

Definition 5.6.4 Let Γ be an individual context, Δ a tag context. Let C be a formula or a place holder denoted by $*$. We define a formulas $Trans(\Gamma, C, \Delta)$ as follows.

$$Trans(\{\}, C, \Delta) = \left\{ \begin{array}{ll} C & \text{if } Dom(\Delta) = \{\} \\ Disj(\Delta) & \text{if } Dom(\Delta) \neq \{\} \text{ and } C = * \\ C \vee Disj(\Delta) & \text{otherwise} \end{array} \right.$$

Let x be the individual variable such that $x \in Dom(\Gamma)$ and $x \leq y$ for any $y \in Dom(\Gamma)$. Then,

$$Trans(\Gamma, C, \Delta) = \begin{cases} \Gamma(x) \supset Trans(\Gamma \backslash x, C, Dep(x, \Delta)) & \text{if } Dom(Indep(x, \Delta)) = \{\} \\ (\Gamma(x) \supset Trans(\Gamma \backslash x, C, Dep(x, \Delta))) & \text{otherwise.} \\ \qquad \vee\; Trans(\Gamma \backslash x, *, Indep(x, \Delta)) & \end{cases}$$

Proposition 5.6.5 $\Gamma \vdash M : C\,;\, \Delta$ *is derivable in* $L_{c/t}$ *if and only if* $Trans(\tilde{\Gamma}, \tilde{C}, \tilde{\Delta})$ *is provable in the propositional fragment of NJ.*

Proof. By induction on the derivation of $\Gamma \vdash M : C\,;\, \Delta$. $\square$

Chapter 6

A term model for the extended system

6.1 Term models

Definition 6.1.1 Let *red* be a subrelation of $\rightarrow$. A *red-sequence* is a possibly infinite sequence $M_0, M_1, M_2, \ldots$ of terms such that M_i *red* M_{i+1} for any i.

Definition 6.1.2 Let *red* be a subrelation of $\rightarrow$. A term M is *strongly normalizable* w.r.t. *red* if there exists no infinite *red*-sequence starting with M. We simply say that M is strongly normalizable if so is w.r.t. $\rightarrow$.

Definition 6.1.3 Let n be a natural number. We define a relation $\xrightarrow{*n}$ inductively as follows.

1. $M \xrightarrow{*0} N$ iff $M = N$.

2. $M \xrightarrow{*n+1} N$ iff $M \xrightarrow{*n} N$ or $M \rightarrow M'$ and $M' \xrightarrow{*n} N$ for some M'.

We also define $\xrightarrow{sub*n}$ in the same way.

Proposition 6.1.4 *Let M be a term, and let x and y be individual variables. Let u and v be tag variables, n a natural number. If $M[y/x, v/u] \xrightarrow{*n} N$, then $M \xrightarrow{*n} N'$ and $N = N'[y/x, v/u]$ for some N'.*

Proof. Straightforward from Proposition 5.3.5. $\square$

Definition 6.1.5 (Types) We define a collection *Type* of sets of terms as follows:

$$Type = \{\, T \mid \text{For any } M \in T,\ M \rightarrow N \text{ implies } N \in T. \,\}\,.$$

Elements of *Type* are called *types*. We use T, S, $\ldots$ to denote types.

Definition 6.1.6 (Regular types) A type T is *regular* if and only if

$M \in T$ implies $M[v/u] \in T$ for any term M and for any tag variables u and v,

Definition 6.1.7 Let V and V' be finite sets of individual variables, and x an individual variable. We define $V[V'/\{x\}]$ as

$$V[V'/\{x\}] \quad = \quad \begin{cases} (V - \{x\}) \cup V' & \text{if } x \in V \\ V & \text{otherwise.} \end{cases}$$

Definition 6.1.8 (Frames) A set of term $\mathcal{X}$ is a *frame* if and only if

1. $\mathcal{X}$ is a regular type,
2. $x \in \mathcal{X}$ for any individual variable x,
3. if $M,\ N \in \mathcal{X}$, then $M[N/x] \in \mathcal{X}$, and
4. if $\Gamma \vdash M : C\,;\, \Delta$ is derivable for some Γ and Δ, then $M \in \mathcal{X}$.

Definition 6.1.9 Let n be a natural number, u a tag variable, T a type, and V a finite set of individual variables. A set of terms $\mathcal{J}_n(u, T, V)$ is defined as follows.

$M \in \mathcal{J}_n(u, T, V)$ if and only if

1. if $M \overset{*n}{\to}$ **throw** $u\ K$, then $K \in T$,
2. if $M \overset{*n}{\to} K$ for some normal form K such that $u \in FTV(K)$ and $FIV(K) \cap V = \{\}$, then $K =$ **throw** $u\ L$ for some closed term L,
3. if $M \overset{*k}{\to}$ **catch** $v\ L$ for some k $(k \leq n)$ and $v \neq u$, then $L \in \mathcal{J}_{n-k}(u, T, V)$,
4. if $M \overset{*k}{\to}$ **throw** $v\ L$ for some k $(k \leq n)$, then $L \in \mathcal{J}_{n-k}(u, T, V)$,
5. if $M \overset{*k}{\to} \lambda\, y.\, L$ for some k $(k \leq n)$, then $L \in \mathcal{J}_{n-k}(u, T, V - \{y\})$,
6. if $M \overset{*k}{\to} L_1\, L_2$ for some k $(k \leq n)$, then $L_1,\ L_2 \in \mathcal{J}_{n-k}(u, T, V)$,
7. if $M \overset{*k}{\to} {<}L_1,\ L_2{>}$ for some k $(k \leq n)$, then $L_1,\ L_2 \in \mathcal{J}_{n-k}(u, T, V)$,
8. if $M \overset{*k}{\to}$ $\mathbf{proj}_i\ L$ for some k $(k \leq n)$ and some i $(i = 1, 2)$, then $L \in \mathcal{J}_{n-k}(u, T, V)$,
9. if $M \overset{*k}{\to}$ $\mathbf{inj}_i\ L$ for some k $(k \leq n)$ and some i $(i = 1, 2)$, then $L \in \mathcal{J}_{n-k}(u, T, V)$,
10. if $M \overset{*k}{\to}$ **case** $L_0\ y_1.L_1\ y_2.L_2$ for some k $(k \leq n)$, then
    - $L_0 \in \mathcal{J}_{n-k}(u, T, V)$, and
    - for any i $(i = 1, 2)$, there exists some V_i' such that $V_i'[FIV(L_0)/\{y_i\}] \subset V$ and $L_i \in \mathcal{J}_{n-k}(u, T, V_i')$,
11. if $M \overset{*k}{\to} \kappa\, v.\, L$ for some k $(k \leq n)$ and $v \neq u$, then $L \in \mathcal{J}_{n-k}(u, T, V)$, and
12. if $M \overset{*k}{\to} L\, v$ for some k $(k \leq n)$, then $L \in \mathcal{J}_{n-k}(u, T, V)$.

Note that $M \in \mathcal{J}_n(u, T, V)$ is defined by induction on the lexicographic ordering of n and $|M|$.

Proposition 6.1.10 *Let M be a term such that $M \in \mathcal{J}_n(u, T, V)$.*

1. *If $m < n$, then $M \in \mathcal{J}_m(u, T, V)$.*
2. *If $M \overset{*k}{\to} N$ for some k $(k \leq n)$, then $N \in \mathcal{J}_{n-k}(u, T, V)$.*

Proof. By induction on the lexicographic ordering of n and $|M|$. $\square$

Proposition 6.1.11 *If $T \subset T'$ and $V \subset V'$, then $\mathcal{J}_n(u, T, V) \subset \mathcal{J}_n(u, T', V')$ for any n and u.*

Proof. By induction on the lexicographic ordering of n and $|M|$. □

Proposition 6.1.12 *If $M \in \mathcal{J}_n(u, T, V)$ and $FIV(M) \subset V'$ then $M \in \mathcal{J}_n(u, T, V')$.*

Proof. By induction on the lexicographic ordering of n and $|M|$. □

Proposition 6.1.13 *If $u \notin FTV(M)$, then $M \in \mathcal{J}_n(u, T, V)$ for any type T.*

Proof. By induction on the lexicographic ordering of n and $|M|$. □

Proposition 6.1.14 *Let T be a regular type. Let M be a term, and v and w tag variables such that $M \in \mathcal{J}_n(u, T, V)$.*

1. *If $w \neq u$, then $M[w/v] \in \mathcal{J}_n(u, T, V)$.*
2. *If $M \in \mathcal{J}_n(v, T, V)$, then $M[w/v] \in \mathcal{J}_n(u, T, V)$.*

Proof. By induction on the lexicographic ordering of n and $|M|$. Use Proposition 6.1.4. □

Definition 6.1.15 Let u be a tag variable, and T a type. We define $\mathcal{J}(u, T, V)$ as follows.

$$\mathcal{J}(u, T, V) = \{M \mid M \in \mathcal{J}_n(u, T, V) \text{ for any } n\}.$$

Definition 6.1.16 Let u be a tag variable, and T a type. We define $\mathcal{J}^-(u, T, V)$ as follows.

- if $M = \textbf{throw}\ u\ K$, then $K \in T$,
- if M is a normal form such that $u \in FTV(M)$ and $FIV(M) \cap V = \{\}$, then $M = \textbf{throw}\ u\ L$ for some closed term L,
- if $M = \textbf{catch}\ v\ L$ and $v \neq u$, then $L \in \mathcal{J}(u, T, V)$,
- if $M = \textbf{throw}\ v\ L$, then $L \in \mathcal{J}(u, T, V)$,
- if $M = \lambda y. L$, then $L \in \mathcal{J}(u, T, V - \{y\})$,
- if $M = L_1\ L_2$, then $L_1, L_2 \in \mathcal{J}(u, T, V)$,
- if $M = <L_1, L_2>$, then $L_1, L_2 \in \mathcal{J}(u, T, V)$,
- if $M = \textbf{proj}_i\ L$ for some i ($i = 1, 2$), then $L \in \mathcal{J}(u, T, V)$,
- if $M = \textbf{inj}_i\ L$ for some i ($i = 1, 2$), then $L \in \mathcal{J}(u, T, V)$,
- $M = \textbf{case}\ L_0\ y_1.L_1\ y_2.L_2$, then
  - $L_0 \in \mathcal{J}(u, T, V)$, and
  - for any i ($i = 1, 2$), there exists some V_i' such that $V_i'[FIV(L_0)/\{y_i\}] \subset V$ and $L_i \in \mathcal{J}(u, T, V_i')$,
- if $M = \kappa\, v. L$ and $v \neq u$, then $L \in \mathcal{J}(u, T, V)$, and

- if $M = L\,v$, then $L \in \mathcal{J}(u, T, V)$.

Proposition 6.1.17 *$M \in \mathcal{J}(u, T, V)$ if and only if*

$$M \xrightarrow{*} N \text{ implies } N \in \mathcal{J}^-(u, T, V).$$

Proof. Obvious from the definitions. □

Proposition 6.1.18 *If $M \in \mathcal{J}(u, T, V)$, then $M \xrightarrow{*} N$ implies $N \in \mathcal{J}(u, T, V)$, that is, $\mathcal{J}(u, T, V)$ is a type.*

Proof. Straightforward from the previous proposition. □

Proposition 6.1.19 *If $u \notin FTV(M)$, then $M \in \mathcal{J}(u, T, V)$ for any T and V.*

Obvious from Proposition 6.1.13 and Proposition 6.1.17.

Proof. □

Proposition 6.1.20 *If $T \subset T'$ and $V \subset V'$ then $\mathcal{J}(u, T, V) \subset \mathcal{J}(u, T', V')$ for any tag variable u.*

Proof. Straightforward from Proposition 6.1.11 and Proposition 6.1.17. □

Proposition 6.1.21 *Let u be a tag variable. If $M \in \mathcal{J}(u, T, V)$ and $FIV(M) \subset V'$ then $M \in \mathcal{J}(u, T, V')$.*

Proof. Straightforward from Proposition 6.1.12 and Proposition 6.1.17. □

Proposition 6.1.22 *Let T be a regular type. Let M be a term, and v and w tag variables such that $M \in \mathcal{J}(u, T, V)$.*

1. *If $w \neq u$, then $M[w/v] \in \mathcal{J}(u, T, V)$.*
2. *If $M \in \mathcal{J}(v, T, V)$, then $M[w/v] \in \mathcal{J}(u, T, V)$.*

Proof. Straight forward from Proposition 6.1.14 and the definition of $\mathcal{J}(u, T, V)$. □

Proposition 6.1.23 *If $M \in \mathcal{J}^-(u, T, V)$ and $M \overset{sub*}{\rightarrow} M'$, then $M' \in \mathcal{J}^-(u, T, V)$.*

Proof. Straightforward from the definition of $\mathcal{J}^-(u, T, V)$ by Proposition 6.1.18. □

Definition 6.1.24 We define a set $\mathcal{D}^-$ of terms as follows. $M \in \mathcal{D}^-$ if and only if

1. for any proper subterm N of M, $N \in \mathcal{D}^-$, and
2. if M is a normal form, then
   (a) if $M = M_1\,M_2$ for some M_1 and M_2, then $FIV(M_1) \neq \{\}$,
   (b) if $M = \mathbf{proj}_i\,M'$ for some i ($i = 1, 2$) and M', then $FIV(M') \neq \{\}$,
   (c) if $M = \mathbf{case}\ M_0\ x.M_1\ y.M_2$ for some M_0, M_1, M_2, x and y, then $FIV(M_0) \neq \{\}$, and
   (d) if $M = M'\,v$ for some M' and v, then $FIV(M') \neq \{\}$.

Definition 6.1.25 Let C be a formula and $\mathcal{X}$ a frame. We define $\mathcal{I}^-(\mathcal{X}, C)$ and $\mathcal{I}(\mathcal{X}, C)$ inductively as follows.

- $M \in \mathcal{I}(\mathcal{X}, C)$ iff
  1. $M \in \mathcal{X}$,
  2. M is strongly normalizable, and
  3. if $M \xrightarrow{*} K$, then $K \in \mathcal{D}^- \cap \mathcal{I}^-(\mathcal{X}, C)$.
- If C is an atomic type expression, then $M \in \mathcal{I}^-(\mathcal{X}, C)$ iff
  1. if M is a normal form and $FIV(M) = \{\}$, then
     - $M =$ **throw** $u\ L$ for some u and L, or
     - $M \in \mathcal{A}(C)$.
- $M \in \mathcal{I}^-(\mathcal{X}, C_1 \supset C_2)$ iff
  1. if $M = \lambda y.\, L$ and $K \in \mathcal{I}(\mathcal{X}, C_1)$, then $L[K/y] \in \mathcal{I}(\mathcal{X}, C_2)$, and
  2. if M is a normal form and $FIV(M) = \{\}$, then
     - $M =$ **throw** $u\ L$ for some u and L, or
     - $M = \lambda y.\, L$ for some y and L.
- $M \in \mathcal{I}^-(\mathcal{X}, C_1 \wedge C_2)$ iff
  1. if $M = <L_1, L_2>$, then $L_1 \in \mathcal{I}(\mathcal{X}, C_1)$ and $L_2 \in \mathcal{I}(\mathcal{X}, C_2)$, and
  2. if M is a normal form and $FIV(M) = \{\}$, then
     - $M =$ **throw** $u\ L$ for some u and L, or
     - $M = <L_1, L_2>$ for some L_1 and L_2.
- $M \in \mathcal{I}^-(\mathcal{X}, C_1 \vee C_2)$ iff
  1. if $M = \mathbf{inj}_i\ L$ for some i $(i = 1, 2)$, then $L \in \mathcal{I}(\mathcal{X}, C_i)$, and
  2. if M is a normal form and $FIV(M) = \{\}$, then
     - $M =$ **throw** $u\ L$ for some u and L, or
     - $M = \mathbf{inj}_i\ L$ for some i $(i = 1, 2)$ and L.
- $M \in \mathcal{I}^-(\mathcal{X}, C_1 \triangleleft C_2)$ iff
  1. if $M = \kappa\, v.\, L$, then $L \in \mathcal{I}(\mathcal{X}, C_1) \cap \mathcal{J}(v, \mathcal{I}(\mathcal{X}, C_2), FIV(L))$, and
  2. if M is a normal form and $FIV(M) = \{\}$, then
     - $M =$ **throw** $u\ L$ for some u and L, or
     - $M = \kappa\, v.\, L$ for some v and L.

Proposition 6.1.26 *Let $\mathcal{X}$ be a frame. $\mathcal{I}(\mathcal{X}, C)$ is a regular type for any formula C.*

Proof. By induction on the structure of C. Use Proposition 5.3.5, Proposition 6.1.22 and the properties of the frame $\mathcal{X}$. □

Proposition 6.1.27 *Let $\mathcal{X}$ be a frame. $x \in \mathcal{I}(\mathcal{X}, C)$ for any type C and any individual variable x.*

Proof. We get $x \in \mathcal{X}$ from the second property of frames. The other requirements for $x \in \mathcal{I}(\mathcal{X}, C)$ are rather trivial since x is an individual variable. $\square$

Lemma 6.1.28 *Let $\mathcal{X}$ be a frame. If* **throw** u $M \in \mathcal{I}(\mathcal{X}, A)$*, then* **throw** u $M \in \mathcal{I}(\mathcal{X}, C)$ *for any C.*

Proof. Suppose that **throw** u $M \in \mathcal{I}(\mathcal{X}, A)$. Obviously **throw** u $M \in \mathcal{X}$ and **throw** u M is strongly normalizable. Suppose that **throw** u $M \xrightarrow{*} K$. By the definition of $\rightarrow$, $K =$ **throw** v L for some v and L. Therefore, $K \in \mathcal{I}^{-}(\mathcal{X}, C)$ for any formula C. $\square$

Lemma 6.1.29 *Let $\mathcal{X}$ be a frame, and C a formula. If*

1. *$M \in \mathcal{X}$,*
2. *$M \overset{sub*}{\rightarrow} K$ implies $K \in \mathcal{D}^{-}$,*
3. *$M \overset{sub*}{\rightarrow} K$ implies $K \in \mathcal{I}^{-}(\mathcal{X}, C)$,*
4. *for any maximal proper subterm N of M, there exists some formula A such that $N \in \mathcal{I}(\mathcal{X}, A)$, and*
5. *if $M \overset{sub*}{\rightarrow} L \underset{\mathrm{n}}{\mapsto} K$ for some L, then $K \in \mathcal{I}(\mathcal{X}, C)$,*

then $M \in \mathcal{I}(\mathcal{X}, C)$.

Proof. First, we show that M is strongly normalizable. By the fourth premise, M is strongly normalizable w.r.t. $\overset{sub}{\rightarrow}$. On the other hand, we get that L is strongly normalizable for any L such that $M \overset{sub*}{\rightarrow} K \mapsto L$ from the last premise. Therefore, so is M. Next, suppose that $M \xrightarrow{*} K$. It is enough to show that $K \in \mathcal{D}^{-} \cap \mathcal{I}^{-}(\mathcal{X}, C)$ since $M \in \mathcal{X}$ by the first premise. There are three possible cases as follows:

1. $M \overset{sub*}{\rightarrow} K$,
2. $M \overset{sub*}{\rightarrow} M'[$**throw** v $L/z] \underset{\mathrm{t}}{\mapsto}$ **throw** v $L \xrightarrow{*} K$ for some M', z, v and L such that $z \in FIV(M')$ and $M' \not\equiv z$.
3. $M \overset{sub*}{\rightarrow} M' \underset{\mathrm{n}}{\mapsto} L \xrightarrow{*} K$ for some M' and L.

In the first case, we get $K \in \mathcal{D}^{-} \cap \mathcal{I}^{-}(\mathcal{X}, C)$ from the second and the third premises. In the second case, there exists some N such that $N \xrightarrow{*}$ **throw** v L and N is a maximal proper subterm of M. Therefore, **throw** v $L \in \mathcal{I}(\mathcal{X}, A)$ for some A by the fourth premise. We get **throw** v $L \in \mathcal{I}(\mathcal{X}, C)$ by Lemma 6.1.28, and therefore, $K \in \mathcal{D}^{-} \cap \mathcal{I}^{-}(\mathcal{X}, C)$ since **throw** v $L \xrightarrow{*} K$. As for the last case, it is obvious because $L \in \mathcal{I}(\mathcal{X}, C)$ from the last premise. This completes the proof of $M \in \mathcal{I}(\mathcal{X}, C)$. $\square$

Lemma 6.1.30 *Let u be a tag variable, and T a type. If*

1. *$M \overset{sub*}{\rightarrow} K$ implies $K \in \mathcal{J}^{-}(u, T, V)$,*

2. $M \overset{sub*}{\to} L \underset{t}{\mapsto} K$ *implies* $K \in \mathcal{J}(u, T, V)$, *and*

3. *if* $M \overset{sub*}{\to} L \underset{n}{\mapsto} K$ *for some* L, *then* $K \in \mathcal{J}(u, T, V)$,

then $M \in \mathcal{J}(u, T, V)$.

Proof. Suppose that $M \overset{*}{\to} K$. There are three possible cases as follows:

1. $M \overset{sub*}{\to} K$,

2. $M \overset{sub*}{\to} M'[\textbf{throw}\ v\ L/z] \underset{t}{\mapsto} \textbf{throw}\ v\ L \overset{*}{\to} K$ for some M', z, v and L such that $z \in FIV(M')$ and $M' \not\equiv z$,

3. $M \overset{sub*}{\to} M' \underset{n}{\mapsto} L \overset{*}{\to} K$ for some M' and L.

It is enough to show that $K \in \mathcal{J}^-(u, T, V)$. In the first case, we get $K \in \mathcal{J}^-(u, T, V)$ from the first premise. In the second case, $K \in \mathcal{J}^-(u, T, V)$ because $\textbf{throw}\ v\ L \in \mathcal{J}(u, T, V)$ by the second premises. As for the third case, $L \in \mathcal{J}(u, T, V)$ from the last premise. Therefore, $K \in \mathcal{J}^-(u, T, V)$. □

Definition 6.1.31 We define a type $\mathcal{D}$ as follows. $M \in \mathcal{D}$ iff

$$M \overset{*}{\to} K \text{ implies } K \in \mathcal{D}^- \text{ for any } K.$$

Definition 6.1.32 A frame $\mathcal{X}$ is *admissible* if and only if

$$\text{if } M, N \in \mathcal{X} \cap \mathcal{J}(u, T, V) \text{ and } MN \in \mathcal{D}, \text{ then } MN \in \mathcal{J}(u, T, V).$$

This property of admissible frames is crucial to the construction of the term model discussed here, but it is not trivial whether such a frame exists or not. We will give an example of admissible frame later.

Theorem 6.1.33 *Let* $\mathcal{X}$ *be an admissible frame. Let* $\Gamma = \{x_1 : A_1, \ldots, x_m : A_m\}$ *and* $\Delta = \{u_1 : B_1, \ldots, u_n : B_n\}$. *Let* $\Gamma \vdash M : C\,;\, \Delta$ *be a derivable judgement. Let* $N_1, \ldots, N_m$ *be terms such that* $N_i \in \mathcal{I}(\mathcal{X}, A_i)$ *for any* i *(*$1 \le i \le m$*). Let* u *a tag variable, and* V *a finite set of individual variables. We define* $\overline{K}$ *and* $\overline{U}$ *as* $\overline{K} = K[N_1/x_1, \ldots, N_m/x_m]$ *and* $\overline{U} = U[FIV(N_1)/\{x_1\}, \ldots, FIV(N_m)/\{x_m\}]$ *for any* K *and* U. *Then,*

- $\overline{M} \in \mathcal{I}(\mathcal{X}, C)$, *and* (6.1)

- *if* $N_i \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), V)$ *for any* i *(*$1 \le i \le m$*), then*

$$\overline{M} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V), \tag{6.2}$$

where $\mathcal{I}(\mathcal{X}, \Delta^t(u)) = \{\}$ *and* $\Delta^v(u) = \{\}$ *if* $u \notin Dom(\Delta)$.

Proof. Let $\Gamma \vdash M : C\,;\, \Delta$ be a derivable judgement. By induction on the depth of the derivation. Suppose that

$$N_i \in \mathcal{I}(\mathcal{X}, A_i) \text{ for any } i\ (1 \le i \le m). \tag{6.3}$$

First of all, we show $\overline{M} \in \mathcal{X}$. Since $\Gamma \vdash M : C\,;\, \Delta$ is derivable, $M \in \mathcal{X}$ by the last property of frames. Since $N_i \in \mathcal{X}$ by (6.3), we get $\overline{M} \in \mathcal{X}$ by the third property of frames. We show the rest properties of $\overline{M}$ for (6.1) and (6.2) by cases according to the form of M.

Case 1: *M is an individual variable.* In this case, we can assume that $M = x_i$ for some i. Therefore, obvious because $\overline{M} = N_i$ and $C = A_i$.

Case 2: $M = \mathbf{catch}\ v\ M'$ *for some v and M'.* That is, $\Gamma \vdash M' : C;\ \Delta \sqcup \{v : C^{V'}\}$ is derivable for some V'. We can assume that $v \notin FTV(N_i)$ for any i. Let Δ' be as $\Delta' = \Delta \sqcup \{v : C^{V'}\}$. By the induction hypothesis,

$$\overline{M'} \in \mathcal{I}(\mathcal{X}, C). \tag{6.4}$$

On the other hand, since $v \notin FTV(N_i)$, we get

$$N_i \in \mathcal{J}(v, \mathcal{I}(\mathcal{X}, \Delta'^t(v)), \{\}) \tag{6.5}$$

for any i by Proposition 6.1.19. Therefore, by the induction hypothesis, we get

$$\overline{M'} \ \in\ \mathcal{J}(v, \mathcal{I}(\mathcal{X}, \Delta'^t(v)), \overline{\Delta'^v(v)}) \ =\ \mathcal{J}(v, \mathcal{I}(\mathcal{X}, C), \overline{\Delta'^v(v)}). \tag{6.6}$$

To show (6.1), we check the five premises of Lemma 6.1.29. We have $\overline{M} \in \mathcal{X}$. For the second premise, suppose that $\overline{M} \overset{sub*}{\rightarrow} K$ for some K. By the definition of $\overset{sub}{\rightarrow}$, we get $K = \mathbf{catch}\ v\ K'$ for some K' such that $\overline{M'} \overset{*}{\rightarrow} K'$. Since $\overline{M'} \in \mathcal{D}$ by (6.4), we get $K' \in \mathcal{D}^-$. Therefore, $K \in \mathcal{D}^-$ from the form of K. For the third premise, suppose that $\overline{M} \overset{sub}{\rightarrow} K = \mathbf{catch}\ v\ K'$ for some K and K' such that $FIV(K) = FIV(K') = \{\}$. We show that K is not a normal form. Since $v \notin FTV(K')$ implies $K \underset{\text{n}}{\mapsto} K'$, we assume that $v \in FTV(K')$. However, by (6.6), $K' = \mathbf{throw}\ v\ L$ for some closed term L. This implies $K = \mathbf{catch}\ v\ \mathbf{throw}\ v\ L \underset{\text{n}}{\mapsto} L$, that is, K is not a normal form. Therefore, third premise is now obvious from the form of $\overline{M}$ and the definition of $\mathcal{I}^-(\mathcal{X}, C)$. On the other hand, the fourth premise holds for $\overline{M}$ by (6.4). As for the last premise, suppose that $\overline{M} \overset{sub*}{\rightarrow} L \underset{\text{n}}{\mapsto} K$ for some L. There are two possible cases as follows:

1. $\overline{M} = \mathbf{catch}\ v\ \overline{M'} \overset{sub*}{\rightarrow} \mathbf{catch}\ v\ K \underset{\text{n}}{\mapsto} K$,
2. $\overline{M} = \mathbf{catch}\ v\ \overline{M'} \overset{sub*}{\rightarrow} \mathbf{catch}\ v\ (\mathbf{throw}\ v\ K) \underset{\text{n}}{\mapsto} K$,

where $v \notin FTV(K)$ in both cases. In the first case, since $\overline{M'} \overset{*}{\rightarrow} K$, we get $K \in \mathcal{I}(\mathcal{X}, C)$ from (6.4). In the second case, since $\overline{M'} \overset{*}{\rightarrow} \mathbf{throw}\ v\ K$, we get $K \in \mathcal{I}(\mathcal{X}, C)$ from (6.6). We now get (6.1) by Lemma 6.1.29.

Next, we show (6.2). Suppose that $N_i \in \mathcal{J}(u, \Delta^t(u), V)$ for any i. Since $v \notin \overline{M}$, it is enough to show that $\overline{M} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta'^t(u)), \overline{\Delta'^v(u)} \cup V)$ by Proposition 6.1.19, and we can assume that $v \neq u$. Since $\Delta^t(u) = \Delta'^t(u)$, by the induction hypothesis,

$$\overline{M'} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta'^t(u)), \overline{\Delta'^v(u)} \cup V). \tag{6.7}$$

We apply Lemma 6.1.30. For the first premise, suppose that $\overline{M'} \overset{sub*}{\rightarrow} K = \mathbf{catch}\ v\ K'$ for some K and K'. We get $K' \in \mathcal{J}^-(u, \mathcal{I}(\mathcal{X}, \Delta'^t(u)), \overline{\Delta'^v(u)} \cup V)$ from (6.7). Suppose that $u \in FTV(K)$ and $FIV(K) \cap (\overline{\Delta'^v(u)} \cup V) = \{\}$. Since $u \in FTV(K')$, $FIV(K') = FIV(K)$ and $\overline{M'} \overset{*}{\rightarrow} K'$, we get $K' = \mathbf{throw}\ u\ L$ for some closed term L from (6.7). That is, K is not a normal form. Therefore, the first premise is obvious from (6.7). For the second premise, suppose that $\overline{M} \overset{sub*}{\rightarrow} L \underset{\text{t}}{\mapsto} K$ for some L. By the definition of $\overset{sub}{\rightarrow}$, we get $\overline{M'} \overset{*}{\rightarrow} K$, and therefore, $K \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta'^t(u)), \overline{\Delta'^v(u)} \cup V)$ from (6.7). To show that the last premise holds, suppose that $\overline{M} \overset{sub*}{\rightarrow} L \underset{\text{n}}{\mapsto} K$ for some L. There are also the two possible cases above. Since $\overline{M'} \overset{*}{\rightarrow} K$ or $\overline{M'} \overset{*}{\rightarrow} \mathbf{throw}\ v\ K$, we get $K \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta'^t(u)), \overline{\Delta'^v(u)} \cup V)$ from (6.7). We now get $\overline{M} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta'^t(u)), \overline{\Delta'^v(u)} \cup V)$ by Lemma 6.1.30.

Case 3: $M = \textbf{throw}\ v\ M'$ *for some* v *and* M'. That is, $\Gamma' \vdash M' : E\,;\ \Delta'$ is derivable for some Γ', E and Δ' such that $\Gamma' \subset \Gamma$ and $\Delta = \Delta' \sqcup \{v : E^{Dom(\Gamma')}\}$. We can assume that $\Delta = \Delta'$ by Proposition 5.4.13. By the induction hypothesis,

$$\overline{M'} \in \mathcal{I}(\mathcal{X},\, E). \tag{6.8}$$

To show (6.1), we check the premises of Lemma 6.1.29. We have $\overline{M} \in \mathcal{X}$. For the second and the third premises, suppose that $\overline{M} \overset{sub*}{\to} K$. We show that $K \in \mathcal{D}^-$ for the second premise. By the definition of $\overset{sub}{\to}$, $K = \textbf{throw}\ v\ K'$ for some K' such that $\overline{M'} \overset{*}{\to} K'$. We get $K' \in \mathcal{D}^-$ from (6.8). For the third premise, $K \in \mathcal{I}^-(\mathcal{X},\, C)$ is obvious from the form of K. We also get the fourth premise from (6.8). As for the last premise, it is impossible that $\overline{M} \overset{sub*}{\to} L \underset{\text{n}}{\mapsto} K$ by the definition of $\overset{sub}{\to}$ and $\underset{\text{n}}{\mapsto}$. We now get (6.1) by Lemma 6.1.29.

Next, we show (6.2). Suppose that $N_i \in \mathcal{J}(u, \mathcal{I}(\mathcal{X},\, \Delta^t(u)),\, V)$ for any i. By the induction hypothesis,

$$\overline{M'} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X},\, \Delta^t(u)),\, \overline{\Delta^t(u)} \cup V). \tag{6.9}$$

We check the three premises of Lemma 6.1.30. For the first premise, suppose that $\overline{M} \overset{sub*}{\to} K = \textbf{throw}\ v\ K'$. If $u = v$, then we get $K' \in \mathcal{I}(\mathcal{X},\, E) = \mathcal{I}(\mathcal{X},\, \Delta^t(v))$ from (6.8) since $\overline{M'} \overset{*}{\to} K'$. On the other hand, we get $K' \in \mathcal{J}(u, \mathcal{I}(\mathcal{X},\, \Delta^t(u)),\, \overline{\Delta^t(u)} \cup V)$. from (6.9). Now, suppose that $u \in FTV(K)$, $FIV(K) \cap (\overline{\Delta^t(u)} \cup V) = \{\}$ and K is a normal form. If $u \in FTV(K')$, then we get $K' = \textbf{throw}\ u\ L$ for some L from (6.9). Therefore, $u \notin FTV(K')$, that is, $u = v$. We get $FIV(K') = \{\}$ from $FIV(K) \cap (\overline{\Delta^t(u)} \cup V) = \{\}$ and

$$FIV(K) \subset FIV(\overline{M'}) \subset \overline{Dom(\Gamma')} \subset \overline{\Delta^t(v)}.$$

Therefore, if $w \in FTV(K')$ for some w, then $K' = \textbf{throw}\ w\ L$ for some L, and this implies that K is not a normal form. That is, K' is a closed term. Thus, the first premise holds. For the second premise, suppose that $\overline{M} \overset{sub*}{\to} L \underset{\text{t}}{\mapsto} K$ for some L. By the definition of $\overset{sub}{\to}$, we get $\overline{M'} \overset{*}{\to} K$, and therefore, $K \in \mathcal{J}(u, \mathcal{I}(\mathcal{X},\, \Delta^t(u)),\, \overline{\Delta^v(u)} \cup V)$ from (6.9). Since it is impossible that $\overline{M} \overset{sub*}{\to} L \underset{\text{n}}{\mapsto} K$, we now get (6.2) by Lemma 6.1.30.

Case 4: $M = \lambda\, y.\, M'$ *for some* y *and* M'. That is, $\Gamma \cup \{y : C_1\} \vdash M' : C_2\,;\ \Delta$ is derivable for some C_1 and C_2 such that $C = C_1 \supset C_2$ and $y \notin \Delta^v(u)$ for any $u \in Dom(\Delta)$. We can assume that $y \notin V$ and $y \notin FIV(N_i)$ for any i. Since $y \in \mathcal{I}(\mathcal{X},\, C_1)$, by the induction hypothesis,

$$\overline{M'} \in \mathcal{I}(\mathcal{X},\, C_2). \tag{6.10}$$

First, we show (6.1). We use Lemma 6.1.29. We have $\overline{M} \in \mathcal{X}$. For the second premise, suppose that $\overline{M} \overset{sub*}{\to} K$ for some K. By the definition of $\overset{sub}{\to}$, we get $K = \lambda\, y.\, K'$ for some K' such that $\overline{M'} \overset{*}{\to} K'$. We get $K' \in \mathcal{D}^-$ from (6.10). Therefore, the second premise holds. As for the third premise, suppose that $\overline{M} = \lambda\, y.\, \overline{M'} \overset{sub*}{\to} \lambda\, y.\, L$. To show that $\lambda\, y.\, L \in \mathcal{I}^-(\mathcal{X},\, C_1 \supset C_2)$, suppose that $K \in \mathcal{I}(\mathcal{X},\, C_1)$. By the induction hypothesis,

$$M'[N_1/x_1, \ldots, N_m/x_m, K/y] \in \mathcal{I}(\mathcal{X},\, C_2).$$

On the other hand, $M'[N_1/x_1, \ldots, N_m/x_m, K/y] = \overline{M'}[K/y] \overset{*}{\to} L[K/y]$. Therefore, $L[K/y] \in \mathcal{I}(\mathcal{X},\, C_2)$. The fourth premise is also satisfied by (6.10). As for the last premise, it is impossible that $\overline{M} \overset{sub*}{\to} L \underset{\text{n}}{\mapsto} K$ by the definition of $\overset{sub}{\to}$ and $\underset{\text{n}}{\mapsto}$. We now get (6.1).

Next, we show (6.2). Suppose that $N_i \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), V)$ for any i. Since $y \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), V)$, by the induction hypothesis,

$$\overline{M'} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V). \tag{6.11}$$

Since $y \notin \Delta^v(u)$, $y \notin V$ and $y \notin FIV(N_i)$ for any i,

$$y \notin \overline{\Delta^v(u)} \cup V. \tag{6.12}$$

We apply Lemma 6.1.30. For the first premise, suppose that $\overline{M} \overset{sub*}{\rightarrow} K = \lambda y. K'$. We get

$$K' \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), (\overline{\Delta^v(u)} \cup V) - \{y\})$$

from (6.11) and (6.12). Suppose that $u \in FTV(K)$ and $FIV(K) \cap (\overline{\Delta^v(u)} \cup V) = \{\}$. Since $FIV(K') \subset FIV(K) \cup \{y\}$, we get $FIV(K') \cap (\overline{\Delta^v(u)} \cup V) = \{\}$ by (6.12). Therefore, $K' =$ **throw** $u\ L$ for some closed term L. This implies that K is not a normal form. Thus, the first premise holds. For the second premise, if $\overline{M} \overset{sub*}{\rightarrow} L \underset{t}{\mapsto} K$ for some L, then we get $\overline{M'} \overset{*}{\rightarrow} K$, and therefore, $K \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V)$ from (6.11). Since the last premise is trivial, we get (6.2) by Lemma 6.1.30.

Case 5: $M = M_1\, M_2$ *for some* M_1 *and* M_2. That is, $\Gamma_1 \vdash M_1 : D \supset C\,;\, \Delta_2$ and $\Gamma_2 \vdash M_2 : D\,;\, \Delta_2$ are derivable for some Γ_1, Γ_2, D, Δ_1 and Δ_2 such that $\Gamma = \Gamma_1 \cup \Gamma_2$ and $\Delta = \Delta_1 \sqcup \Delta_2$. By the induction hypothesis,

$$\overline{M_1} \in \mathcal{I}(\mathcal{X}, D \supset C), \tag{6.13}$$
$$\overline{M_2} \in \mathcal{I}(\mathcal{X}, D). \tag{6.14}$$

First, we show (6.1) by applying Lemma 6.1.29. We have $\overline{M} \in \mathcal{X}$. For the second premise, suppose that $\overline{M} \overset{sub*}{\rightarrow} K$ for some K. By the definition of $\overset{sub}{\rightarrow}$, we get $K = K_1\, K_2$ for some K_1 and K_2 such that $\overline{M_1} \overset{*}{\rightarrow} K_1$ and $\overline{M_2} \overset{*}{\rightarrow} K_2$. We get $K_1, K_2 \in \mathcal{D}^-$ from (6.13) and (6.14). Suppose that K is a normal form. We get $FIV(K_1) \neq \{\}$ since $FIV(K_1) = \{\}$ implies that

- $K_1 =$ **throw** $v\ L$ and $K \underset{t}{\mapsto}$ **throw** $v\ L$ for some v and L, or
- $K_1 = \lambda y. L$ and $K \underset{n}{\mapsto} L[K_2/y]$ for some y and L,

by (6.13). Therefore, $K \in \mathcal{D}^-$. For the third premise, we similarly get that K is not a normal form if $\overline{M} \overset{sub*}{\rightarrow} K$ and $FIV(K) = \{\}$. The fourth premise is obvious from (6.13) and (6.14). As for the last premise, suppose that $\overline{M} \overset{sub*}{\rightarrow} L \underset{n}{\mapsto} K$, that is, for some y, L_1 and L_2,

$$\overline{M_1}\, \overline{M_2} \overset{sub*}{\rightarrow} (\lambda y. L_1)\, L_2 \underset{n}{\mapsto} L_1[L_2/y] = K.$$

Therefore, $K \in \mathcal{I}(\mathcal{X}, C)$ from (6.13) and (6.14). We now get

$$\overline{M_1}\, \overline{M_2} \in \mathcal{I}(\mathcal{X}, C) \tag{6.15}$$

by Lemma 6.1.29.

Now, we show (6.2). Suppose that $N_i \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), V)$ for any i. By the induction hypothesis, $\overline{M_1} \in \mathcal{J}(C, \mathcal{I}(\mathcal{X}, \Delta_1^t(u)), \overline{\Delta_1^v(u)} \cup V)$ and $\overline{M_2} \in \mathcal{J}(C, \mathcal{I}(\mathcal{X}, \Delta_2^t(u)), \overline{\Delta_2^v(u)} \cup V)$. Therefore, we get

$$\overline{M_1},\ \overline{M_2} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V).$$

Since $\mathcal{X}$ is admissible, we get $\overline{M} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V)$ from (6.13), (6.14) and (6.15).

Case 6: $M = \kappa v.M'$ *for some* v *and* M'. That is, $\Gamma \vdash M' : C_1\,;\,\Delta \sqcup \{v : C_2^{V'}\}$ is derivable for some C_1, C_2 and V' such that $C = C_1 \triangleleft C_2$. We can assume that $v \notin FTV(N_i)$ for any i. Let Δ' be as $\Delta' = \Delta \sqcup \{v : C_2^{V'}\}$. By the induction hypothesis,

$$\overline{M'} \in \mathcal{I}(\mathcal{X}, C_1). \tag{6.16}$$

To show (6.1), we check the premises of Lemma 6.1.29. We have $\overline{M} \in \mathcal{X}$. For the second premise, suppose that $\overline{M} \xrightarrow{sub*} K$ for some K. By the definition of $\xrightarrow{sub}$, we get $K = \kappa v.K'$ for some K' such that $\overline{M'} \xrightarrow{*} K'$. We get $K' \in \mathcal{D}^-$ from (6.16). Therefore, the second premise holds. For the third premise, suppose that $\overline{M} = \kappa v.\overline{M'} \xrightarrow{sub*} \kappa v.L$. We show that $\kappa v.L \in \mathcal{I}^-(\mathcal{X}, C_1 \triangleleft C_2)$. Since $\overline{M'} \xrightarrow{*} L$, we get $L \in \mathcal{I}(\mathcal{X}, C_1)$ from (6.16). On the other hand, we get $N_i \in \mathcal{J}(v, \mathcal{I}(\mathcal{X}, C_2), \{\})$ since $v \notin FTV(N_i)$ for any i. Therefore, by the induction hypothesis,

$$\overline{M'} \in \mathcal{J}(v, \mathcal{I}(\mathcal{X}, C_2), \overline{\Delta'^v(v)}).$$

Since $\overline{M'} \xrightarrow{*} L$, we get $L \in \mathcal{J}(v, \mathcal{I}(\mathcal{X}, C_2), \overline{\Delta'^v(v)})$, and therefore, $L \in \mathcal{J}(v, \mathcal{I}(\mathcal{X}, C_2), FIV(L))$ by Proposition 6.1.21. As for the last premise, it is impossible that $\overline{M} \xrightarrow{sub*} L \underset{\mathrm{n}}{\hookrightarrow} K$ by the definition of $\xrightarrow{sub}$ and $\underset{\mathrm{n}}{\hookrightarrow}$. We now get (6.1) by Lemma 6.1.29.

Next, we show (6.2). Suppose that $N_i \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), V)$ for any i. Since $v \notin \overline{M}$, it is enough to show that $\overline{M} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta'^t(u)), \overline{\Delta'^v(u)} \cup V)$ by Proposition 6.1.19, and we can assume that $v \neq u$. Since $\Delta^t(u) = \Delta'^t(u)$, by the induction hypothesis,

$$\overline{M'} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta'^t(u)), \overline{\Delta'^v(u)} \cup V). \tag{6.17}$$

We apply Lemma 6.1.30. For the first premise, suppose that $\overline{M} \xrightarrow{sub*} K = \kappa v.K'$ for some K and K'. We get $K' \in \mathcal{J}^-(u, \mathcal{I}(\mathcal{X}, \Delta'^t(u)), \overline{\Delta'^v(u)} \cup V)$ from (6.17). Suppose that $u \in FTV(K)$ and $FIV(K) \cap (\overline{\Delta'^v(u)} \cup V) = \{\}$. Since $u \in FTV(K')$, $FIV(K') = FIV(K)$ and $\overline{M'} \xrightarrow{*} K'$, we get $K' = \textbf{throw}\ u\ L$ for some closed term L from (6.17). Therefore, K is not a normal form. Thus, the first premise holds. For the second premise, if $\overline{M} \xrightarrow{sub*} L \underset{\mathrm{t}}{\hookrightarrow} K$ for some L, then we get $K \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta'^t(u)), \overline{\Delta'^v(u)} \cup V)$ from (6.17) since $\overline{M'} \xrightarrow{*} K$. Since the last premise is trivial, we get (6.2) by Lemma 6.1.30.

Case 7: $M = M'\,v$ *for some* M' *and* v. That is, $\Gamma' \vdash M' : C \triangleleft E\,;\,\Delta'$ is derivable for some Γ', E and Δ' such that $\Gamma' \subset \Gamma$ and $\Delta = \Delta' \sqcup \{v : E^{Dom(\Gamma')}\}$. We can assume that $\Delta = \Delta'$ by Proposition 5.4.13. By the induction hypothesis,

$$\overline{M'} \in \mathcal{I}(\mathcal{X}, C \triangleleft E). \tag{6.18}$$

To show (6.1), we check the premises of Lemma 6.1.29. We have $\overline{M} \in \mathcal{X}$. For the second premise, suppose that $\overline{M} \xrightarrow{sub*} K$ for some K. We show that $K \in \mathcal{D}^-$. By the definition of $\xrightarrow{sub}$, we get $K = K'\,v$ for some K' such that $\overline{M'} \xrightarrow{*} K'$. We get $K' \in \mathcal{D}^-$ from (6.18). If $FIV(K) = \{\}$, then we get $K' = \textbf{throw}\ v'\ L$ or $K' = \kappa v'.L$ for some v' and L from (6.18), that is, K is not a normal form. Thus, we get $K \in \mathcal{D}^-$. For the third premise, we similarly get that K is not a normal form if $\overline{M} \xrightarrow{sub*} K$ and $FIV(K) = \{\}$. The fourth premise is obvious from (6.18). As for the last premise, suppose that $\overline{M} \xrightarrow{sub*} L \underset{\mathrm{n}}{\hookrightarrow} K$, that is, for some v' and L',

$$\overline{M'}\,v \xrightarrow{sub*} (\kappa v'.L')\,v \underset{\mathrm{n}}{\hookrightarrow} L'[v/v'] = K.$$

We get $L' \in \mathcal{I}(\mathcal{X}, C)$ from (6.18) by the definition of $\mathcal{I}(\mathcal{X}, C \triangleleft E)$, and therefore, $K \in \mathcal{I}(\mathcal{X}, C)$ since $\mathcal{I}(\mathcal{X}, C)$ is a regular type. We now get $\overline{M} \in \mathcal{I}(\mathcal{X}, C)$ by Lemma 6.1.29.

Next, we show (6.2). Suppose that $N_i \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), V)$ for any i. By the induction hypothesis,

$$\overline{M'} \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V). \tag{6.19}$$

We apply Lemma 6.1.30. For the first premise, suppose that $\overline{M} \overset{sub*}{\rightarrow} K = K'\,v$. We get $K' \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V)$ from (6.19). Suppose that $u \in FTV(K)$ and $FIV(K) \cap (\overline{\Delta^v(u)} \cup V) = \{\}$. We show that K is not a normal form. Since $FIV(K') = FIV(K)$, we get

$$FIV(K') \cap (\overline{\Delta^v(u)} \cup V) = \{\}.$$

If $u \in FTV(K')$, then $K' = \textbf{throw}\ u\ L$ for some closed term L from (6.19), and this implies that K is not a normal form. On the other hand, if $u = v$, then $FIV(K') = \{\}$ since $FIV(K) \subset FIV(\overline{M'}) \subset \overline{Dom(\Gamma')} \subset \overline{\Delta^v(v)}$. Therefore, if K' is a normal from, then $K' = \textbf{throw}\ w\ L$ or $K' = \kappa\, w.\, L$ for some w and L by (6.18). This implies that K is not a normal form. Thus, the first premise holds. For the second premise, if $\overline{M} \overset{sub*}{\rightarrow} L \underset{\mathrm{t}}{\mapsto} K$ for some L, then we get $\overline{M'} \overset{*}{\rightarrow} K$, and therefore, $K \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V)$ from (6.19). As for the last premise, suppose that $\overline{M} \overset{sub*}{\rightarrow} L \underset{\mathrm{n}}{\mapsto} K$ for some L and K, that is, $\overline{M} \overset{sub*}{\rightarrow} (\kappa\, v'.\, L')\, v \underset{\mathrm{n}}{\mapsto} L'[v/v'] = K$ for some v' and L'. We get

$$L' \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V) \tag{6.20}$$

from (6.19). If $v \neq u$, then $L'[v/v'] \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V)$ by Proposition 6.1.22. Therefore, we assume that $v = u$. We have $L' \in \mathcal{J}(v, \mathcal{I}(\mathcal{X}, E), FIV(L'),)$ by (6.18). Since $\Delta^t(u) = \Delta^t(v) = E$ and $FIV(L') \subset FIV(\overline{M}) \subset \overline{\Delta^v(v)} = \overline{\Delta^v(u)}$, we get

$$L' \in \mathcal{J}(v, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V)$$

by Proposition 6.1.20. Therefore, $L'[v/v'] \in \mathcal{J}(u, \mathcal{I}(\mathcal{X}, \Delta^t(u)), \overline{\Delta^v(u)} \cup V)$ from (6.20) by Proposition 6.1.22. We now get (6.2) by Lemma 6.1.30.

Case 8: *M has one of the other forms.* We similarly get (6.1) and (6.2) by Lemma 6.1.29 and Lemma 6.1.30, respectively. □

6.2 An admissible frame

Definition 6.2.1 A *tag dependency* is a finite mapping which assigns a set of individual variables to each tag variable in its domain. We use $\delta, \delta', \ldots$ to denote tag dependencies. Let $u_1, \ldots, u_n$ be tag variables, and let $V_1, \ldots, V_n$ be sets of individual variables such that if $u_i = u_j$ then $V_i = V_j$ for any i and j. We use $\{u_1 : V_1, \ldots, u_n : V_n\}$ to denote a tag dependency whose domain is $\{u_1, \ldots, u_n\}$ and which assigns V_i to u_i for each i.

Definition 6.2.2 Let δ and δ' be tag dependencies. We define a tag dependency $\delta \sqcup \delta'$ as follows.

$$\begin{array}{rcl} Dom(\delta \sqcup \delta') & = & Dom(\delta) \cup Dom(\delta') \\ (\delta \sqcup \delta')(u) & = & \left\{ \begin{array}{ll} \delta(u) \cup \delta'(u) & \text{if } u \in Dom(\delta) \cap Dom(\delta') \\ \delta(u) & \text{if } u \in Dom(\delta) \text{ and } u \notin Dom(\delta') \\ \delta'(u) & \text{if } u \notin Dom(\delta) \text{ and } u \in Dom(\delta') \end{array} \right. \end{array}$$

Definition 6.2.3 Let M be a term. Let U and δ be a set of individual variables and a tag dependency, respectively, where $\delta(u) \subset U$ for any $u \in Dom(\delta)$. *Frame judgements* have the following form.

$$U \vdash_f M \,;\, \delta$$

Definition 6.2.4 We define a formal system for frame judgements.

$$\frac{}{U \cup \{x\} \vdash_f x \,;\, \delta}\ (var) \qquad\qquad \frac{}{U \vdash_f c \,;\, \delta}\ (const)$$

$$\frac{U \vdash_f M \,;\, \delta \sqcup \{u\!:\!V\}}{U \vdash_f \mathbf{catch}\ u\ M \,;\, \delta}\ (catch) \qquad\qquad \frac{U_1 \vdash_f M \,;\, \delta}{U_1 \cup U_2 \vdash_f \mathbf{throw}\ u\ M \,;\, \delta \sqcup \{u\!:\!U_1\}}\ (throw)$$

$$\frac{U \cup \{x\} \vdash_f M \,;\, \delta}{U \vdash_f \lambda x.\, M \,;\, \delta}\ (\supset\text{-I}) \quad (x \notin \delta(u) \text{ for any } u \in Dom(\delta))$$

$$\frac{U_1 \vdash_f M \,;\, \delta_1 \quad U_2 \vdash_f N \,;\, \delta_2}{U_1 \cup U_2 \vdash_f M\, N \,;\, \delta_1 \sqcup \delta_2}\ (\supset\text{-E})$$

$$\frac{U \vdash_f M \,;\, \delta \sqcup \{u\!:\!V\}}{U \vdash_f \kappa\, u.\, M \,;\, \delta}\ (\triangleleft\text{-I}) \qquad \frac{U_1 \vdash_f M \,;\, \delta}{U_1 \cup U_2 \vdash_f M\, u \,;\, \delta \sqcup \{u\!:\!U_1\}}\ (\triangleleft\text{-E})$$

$$\frac{U_1 \vdash_f M \,;\, \delta_1 \quad U_2 \vdash_f N \,;\, \delta_2}{U_1 \cup U_2 \vdash_f \,<\!M,\ N\!>\, ;\, \delta_1 \sqcup \delta_2}\ (\wedge\text{-I})$$

$$\frac{U \vdash_f M \,;\, \delta}{U \vdash_f \mathbf{proj_1}\, M \,;\, \delta}\ (\wedge_1\text{-E}) \qquad\qquad \frac{U \vdash_f M \,;\, \delta}{U \vdash_f \mathbf{proj_2}\, M \,;\, \delta}\ (\wedge_2\text{-E})$$

$$\frac{U \vdash_f M \,;\, \delta}{U \vdash_f \mathbf{inj_1}\, M \,;\, \delta}\ (\vee_1\text{-I}) \qquad\qquad \frac{U \vdash_f M \,;\, \delta}{U \vdash_f \mathbf{inj_2}\, M \,;\, \delta}\ (\vee_2\text{-I})$$

$$\frac{U_1 \vdash_f L \,;\, \delta_1 \quad U_2 \cup \{x\} \vdash_f M \,;\, \delta_2 \quad U_3 \cup \{y\} \vdash_f N \,;\, \delta_3}{U_1 \cup U_2 \cup U_3 \vdash_f \mathbf{case}\ L\ x.M\ y.N \,;\, \delta_1 \sqcup \delta_2[U_1/\{x\}] \sqcup \delta_3[U_1/\{y\}]}\ (\vee\text{-E})$$

We presents a some basic properties of this formal system.

Proposition 6.2.5 *Let M be a term. Let Γ be an individual context, C a formula, and Δ a tag context. If $\Gamma \vdash M : C \,;\, \Delta$ is derivable, then $Dom(\Gamma) \vdash_f M \,;\, \Delta^v$ is derivable.*

Proof. Straightforward induction on the derivation of $\Gamma \vdash M : C \,;\, \Delta$. □

Proposition 6.2.6 *If $U \vdash_f M \,;\, \delta$ is derivable, then $FIV(M) \subset U$ and $FTV(M) \subset Dom(\delta)$.*

Proof. By induction on the derivation of $U \vdash_f M \,;\, \delta$. $\square$

Definition 6.2.7 Let δ and δ' be tag dependencies. We define a relation $\delta \sqsubset \delta'$ as follows. The relation $\delta \sqsubset \delta'$ holds if and only if

- $Dom(\delta) \subset Dom(\delta')$, and
- $\delta(u) \subset \delta'(u)$ for any $u \in Dom(\delta)$.

Proposition 6.2.8 *Let $U \vdash_f M \,;\, \delta$ be a d-derivable frame judgement.*

1. *If $U \subset U'$ and $\delta \sqsubset \delta'$, then $U' \vdash_f M \,;\, \delta'$ is also d-derivable.*
2. *$U[\{y\}/\{x\}] \vdash_f M[y/x] \,;\, \delta[\{y\}/\{x\}]$ is also d-derivable.*
3. *If $\delta[v/u]$ is well defined, then $U \vdash_f M[v/u] \,;\, \delta[v/u]$ is also d-derivable.*

Proof. By simultaneous inductions on d. $\square$

Proposition 6.2.9 *Let x and u be as $x \notin FIV(M)$ and $u \notin FTV(M)$.*

1. *If $U \cup \{x\} \vdash_f M \,;\, \delta$ is derivable, then $U \vdash_f M \,;\, \delta[\{\}/\{x\}]$ is also derivable.*
2. *If $U \vdash_f M \,;\, \delta \sqcup \{u : V\}$ is derivable, then $U \vdash_f M \,;\, \delta$ is also derivable.*

Proof. Straightforward induction on the derivations. $\square$

Proposition 6.2.10 *If $U_1 \vdash_f N \,;\, \delta_1$ and $U_2 \cup \{x\} \vdash_f M \,;\, \delta_2$ are derivable, then $U_1 \cup U_2 \vdash_f M[N/x] \,;\, \delta_1 \sqcup \delta_2[U_1/\{x\}]$ is also derivable.*

Proof. By induction on the depth of the derivation of $U_2 \cup \{x\} \vdash_f M \,;\, \delta_2$. Suppose that $U_1 \vdash_f N \,;\, \delta_1$ and $U_2 \cup \{x\} \vdash_f M \,;\, \delta_2$ are derivable. First, suppose also that $x \notin FIV(M)$, that is, $M[N/x] = M$. Since $U_2 \cup \{x\} \vdash_f M \,;\, \delta_2$ is derivable, so is $U_2 \vdash_f M \,;\, \delta_2[\{\}/\{x\}]$ by Proposition 6.2.9, and this implies that $U_1 \cup U_2 \vdash_f M \,;\, \delta_1 \sqcup \delta_2[U_1/\{x\}]$ is derivable by Proposition 6.2.8. Therefore, we now assume that

$$x \in FIV(M). \tag{6.21}$$

By cases according to the form of M.

Case 1: *$M = y$ for some individual variable y.* In this case, $y \in U_2 \cup \{x\}$. If $M = x$, then we can derive $U_1 \cup U_2 \vdash_f M[N/x] \,;\, \delta_1 \sqcup \delta_2[Dom(U_1)/\{x\}]$ by applying Proposition 6.2.8 to the derivation of $U_1 \vdash_f N \,;\, \delta_1$ since $M[N/x] = N$ and $C = A$ in this case. If $M \neq x$, then we can derive it by (var) since $M[N/x] = y$ and $y \in U_2$ in this case.

Case 2: *$M =$ **catch** $u\ M'$ for some u and M'.* In this case, the following judgement is derivable for some V.

$$U_2 \cup \{x\} \vdash_f M' \,;\, \delta_2 \sqcup \{u : V\}$$

We can assume that $u \notin Dom(\delta_1)$ by Proposition 6.2.8. By the induction hypothesis, we have a derivation of

$$U_1 \cup U_2 \vdash_f M'[N/x] \,;\, \delta_1 \sqcup (\delta_2 \sqcup \{u : V\})[Dom(U_1)/\{x\}]. \tag{6.22}$$

Since $u \notin Dom(\delta_1)$, we get $M[N/x] = \textbf{catch}\ u\ (M'[N/x])$. By applying $(catch)$ to (6.22), we get $U_1 \cup U_2 \vdash_f M[N/x] \,;\, \delta_1 \sqcup \delta_2[Dom(U_1)/\{x\}]$.

Case 3: $M = \textbf{throw}\ u\ M'$ *for some* u *and* M'. In this case, the following judgement is derivable for some U_2' and δ such that $U_2' \subset U_2 \cup \{x\}$ and $\delta_2 = \delta \sqcup \{u : U_2'\}$.

$$U_2' \vdash_f M' ; \delta$$

We get $x \in FIV(M')$ from (6.21), that is, $x \in U_2'$. Let U be as $U = U_2' - \{x\}$. Note that $U_2' = U \cup \{x\}$. Therefore, by Proposition 6.2.8,

$$U \cup \{x\} \vdash_f M' ; \delta.$$

By the induction hypothesis, we have a derivation of

$$U_1 \cup U \vdash_f M'[N/x] ; \delta_1 \sqcup \delta[U_1/\{x\}].$$

Since $M[N/x] = \textbf{throw}\ u\ (M'[N/x])$, by applying $(throw)$,

$$U_1 \cup U \vdash_f M[N/x] ; \delta_1 \sqcup \delta[U_1/\{x\}] \sqcup \{u : U_1 \cup U\}.$$

Since $U \subset U_2$, by Proposition 6.2.8 again,

$$U_1 \cup U_2 \vdash_f M[N/x] ; \delta_1 \sqcup \delta[U_1/\{x\}] \sqcup \{u : U_1 \cup U\}.$$

Note that $\delta[U_1/\{x\}] \sqcup \{u : U_1 \cup U\} = \delta_2[U_1/\{x\}]$ because $\delta_2 = \delta \sqcup \{u : U \cup \{x\}\}$ and $x \notin U$.

Case 4: $M = \lambda y. M'$ *for some* y *and* M'. In this case, $y \notin \delta_2(u)$ for any $u \in Dom(\delta_2)$ and the following judgement is derivable.

$$U_2 \cup \{x\} \cup \{y\} \vdash_f M' ; \delta_2$$

We can assume that $y \notin U_1$ by Proposition 6.2.8, and get $M'[N/x] = \lambda y. (M[N/x])$. By the induction hypothesis, we have a derivation of

$$U_1 \cup U_2 \cup \{y\} \vdash_f M'[N/x] ; \delta_1 \sqcup \delta_2[U_1/\{x\}]. \tag{6.23}$$

Since $y \notin \delta_2(u)$ for any $u \in Dom(\delta_2)$ and $y \notin U_1$, we get $y \notin (\delta_1 \sqcup \delta_2[U_1/\{x\}])(u)$ for any $u \in Dom(\delta_1 \sqcup \delta_2[U_1/\{x\}])$. Therefore, we can derive $U_1 \cup U_2 \vdash_f \lambda y. (M'[N/x]) ; \delta_1 \sqcup \delta_2[U_1/\{x\}]$ by applying ($\supset$-I) to (6.23).

Case 5: *M has one of the other forms.* Similarly, $U_1 \cup U_2 \vdash_f M[N/x] ; \delta_1 \sqcup \delta_2[U_1/\{x\}]$ is derivable. □

Lemma 6.2.11 *If $U \vdash_f M ; \delta$ is derivable and $M \underset{t}{\mapsto}$* **throw** *$v\ N$, then $U \vdash_f$* **throw** *$v\ N ; \delta$ is also derivable.*

Proof. By induction on the depth of the derivation of $U \vdash_f M ; \delta$. Suppose that $U \vdash_f M ; \delta$ is derivable and $M \underset{t}{\mapsto} \textbf{throw}\ v\ N$. By cases according to the form of M.

Case 1: $M = x$ *for some individual variable* x. This is impossible because $M \underset{t}{\mapsto} \textbf{throw}\ v\ N$.

Case 2: $M = \textbf{catch}\ u\ M'$ *for some* u *and* M'*.* In this case, the following judgement is derivable for some V.

$$U \vdash_f M' \,;\, \delta \sqcup \{u : V\} \tag{6.24}$$

We get $u \notin FTV(\textbf{throw}\ v\ N)$ and

$$M' = \textbf{throw}\ v\ N \quad \text{or} \quad M' \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$$

from $M \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$. Therefore, from (6.24) or the induction hypothesis on (6.24),

$$U \vdash_f \textbf{throw}\ v\ N \,;\, \delta \sqcup \{u : V\}.$$

We get $U \vdash_f \textbf{throw}\ v\ N \,;\, \delta$ by Proposition 6.2.9 since $u \notin FTV(\textbf{throw}\ v\ N)$.

Case 3: $M = \textbf{throw}\ u\ M'$ *for some* u *and* M'*.* In this case, the following judgement is derivable for some U' and δ' such that $U' \subset U$ and $\delta = \delta' \sqcup \{u : U'\}$.

$$U' \vdash_f M' \,;\, \delta' \tag{6.25}$$

We get $M' = \textbf{throw}\ v\ N$ or $M' \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$ from $M \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$. Therefore, from (6.25) or the induction hypothesis on (6.25),

$$U' \vdash_f \textbf{throw}\ v\ N \,;\, \delta'.$$

We get $U \vdash_f \textbf{throw}\ v\ N \,;\, \delta$ by Proposition 6.2.8 since $U' \subset U$ and $\delta' \sqsubset \delta$.

Case 4: $M = \lambda\, x.\, M'$ *for some* x *and* M'*.* In this case, $x \notin \delta(u)$ for any $u \in Dom(\delta)$ and the following judgement is derivable.

$$U \cup \{x\} \vdash_f M' \,;\, \delta \tag{6.26}$$

We get $x \notin FIV(\textbf{throw}\ v\ N)$ and

$$M' = \textbf{throw}\ v\ N \quad \text{or} \quad M' \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$$

from $M \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ v\ N$. Therefore, from (6.26) or the induction hypothesis on (6.26),

$$U \cup \{x\} \vdash_f \textbf{throw}\ v\ N \,;\, \delta.$$

We get $U \vdash_f \textbf{throw}\ v\ N \,;\, \delta$ by Proposition 6.2.9 since $x \notin FIV(\textbf{throw}\ v\ N)$.

Case 5: *M has one of the other forms.* Similarly, $U \vdash_f \textbf{throw}\ v\ N \,;\, \delta$ is derivable. □

Lemma 6.2.12 *If $U \vdash_f M \,;\, \delta$ is derivable and $M \underset{\mathrm{n}}{\mapsto} N$, then $U \vdash_f N \,;\, \delta$ is also derivable.*

Proof. By induction on the depth of the derivation of $U \vdash_f M \,;\, \delta$. Suppose that $U \vdash_f M \,;\, \delta$ is derivable and $M \underset{\mathrm{n}}{\mapsto} N$. By cases according to the form of M.

Case 1: $M = \textbf{catch}\ u\ N$ *and* $u \notin FTV(N)$*.* In this case, $U \vdash_f N \,;\, \delta \sqcup \{u : V\}$ is derivable for some V. We get $U \vdash_f N \,;\, \delta$ by Proposition 6.2.9 since $u \notin FTV(N)$.

Case 2: *$M = \textbf{catch}\ u\ (\textbf{throw}\ u\ N)$ and $u \notin FTV(N)$.* The following judgement is derivable for some V, U' and δ' such that $U' \subset U$ and $\delta \sqcup \{u : V\} = \delta' \sqcup \{u : U'\}$.

$$U' \vdash_f N ; \delta'$$

Since $U' \subset U$ and $\delta' \sqsubset \delta \sqcup \{u : V\}$, $U \vdash_f N ; \delta \sqcup \{u : V\}$ is derivable by Proposition 6.2.8. Therefore, $U \vdash_f N ; \delta$ is also derivable by Proposition 6.2.9 since $u \notin FTV(N)$.

Case 3: *$M = (\lambda x. M_1) M_2$ and $N = M_1[M_2/x]$ for some x, M_1 and M_2.* The following two judgements are derivable for some U_1, U_2, δ_1 and δ_2 such that $U = U_1 \cup U_2$, $\delta = \delta_1 \sqcup \delta_2$ and $x \notin \delta_1(u)$ for any $u \in Dom(\delta)$.

$$U_1 \cup \{x\} \vdash_f M_1 ; \delta_1 \tag{6.27}$$

$$U_2 \vdash_f M_2 ; \delta_2 \tag{6.28}$$

We get $U \vdash_f M_1[M_2/x] ; \delta_2 \sqcup \delta_1[U_2/\{x\}]$ from (6.27) and (6.28) by Proposition 6.2.10, where $\delta_2 \sqcup \delta_1[U_2/\{x\}] = \delta$ since $x \notin \delta_1(u)$ for any $u \in Dom(\delta_1)$.

Case 4: *$M = (\kappa u. M') v$ and $N = M'[v/u]$ for some u, v and M'.* The following judgement is derivable for some U', δ' and V such that $U' \subset U$ and $\delta = \delta' \sqcup \{v : U'\}$.

$$U' \vdash_f M' ; \delta' \sqcup \{u : V\}$$

By Proposition 6.2.8, $U' \vdash_f M'[v/u] ; \delta'[v/u] \sqcup \{v : V\}$ is derivable, and since $U' \subset U$, by Proposition 6.2.8 again,

$$U \vdash_f M'[v/u] ; \delta'[v/u] \sqcup \{v : V\}.$$

Since $V \subset U'$ and $\delta'(u) \subset U$,

$$\delta'[v/u] \sqcup \{v : V\} \sqsubset \delta'[v/u] \sqcup \{v : U'\} \sqsubset \delta' \sqcup \{v : U'\} = \delta.$$

Therefore, $U \vdash_f M'[v/u] ; \delta$ is derivable by Proposition 6.2.8.

Case 5: *M has one of the other forms.* Similar. □

Lemma 6.2.13 *If $U \vdash_f M ; \delta$ is derivable and $M \mapsto N$, then $U \vdash_f N ; \delta$ is also derivable.*

Proof. Straightforward from Lemma 6.2.11 and Lemma 6.2.12. □

Proposition 6.2.14 *If $U \vdash_f M ; \delta$ is derivable and $M \to N$, then $U \vdash_f N ; \delta$ is also derivable.*

Proof. By induction on the depth of the derivation of $U \vdash_f M ; \delta$. Suppose that $U \vdash_f M ; \delta$ is derivable and $M \to N$. If $M \mapsto N$, then trivial by Lemma 6.2.13. Therefore we can assume that $M \to N$ and $M \not\mapsto N$. By cases according to the form of M. A typical one is the case that $M = \textbf{throw}\ u\ M'$ for some u and M'. In this case,

$$U' \vdash_f M' ; \delta'$$

is derivable for some U' and δ' such that $U' \subset U$ and $\delta = \delta' \sqcup \{u : U'\}$. Since $M \to N$ and $M \not\mapsto N$, $M' \to N'$ and $N = \textbf{throw}\ u\ N'$ for some N'. Therefore, $U' \vdash_f N' ; \delta'$ is derivable by the induction hypothesis. We get $U \vdash_f \textbf{throw}\ u\ N' ; \delta$ by applying (*throw*). The proofs for other cases are just similar. □

Definition 6.2.15 We define a set of term $\mathcal{X}_0$ as follows:

$$\mathcal{X}_0 = \{\ M \mid U \vdash_f M \,;\, \delta \text{ is derivable for some } U \text{ and } \delta\}.$$

Proposition 6.2.16 *$\mathcal{X}_0$ is a frame.*

Proof. $\mathcal{X}_0$ is a type by Proposition 6.2.14, and is regular by Proposition 6.2.8. $x \in \mathcal{X}_0$ for any x, since $\{x\} \vdash_f x \,;\, \{\}$ is derivable. On the other hand, $M, N \in \mathcal{X}_0$ implies $M[N/x] \in \mathcal{X}_0$ by Proposition 6.2.10. Finally, by Proposition 6.2.5, we get $M \in \mathcal{X}_0$ for any term M such that $\Gamma \vdash M : C \,;\, \Delta$ is derivable for some Γ, C and Δ. $\square$

Definition 6.2.17 Let n be a natural number, u a tag variable, and T a type. We define $\mathcal{J}_n^-(u, T, V)$ as follows.

- if $M = \mathbf{throw}\ u\ K$, then $K \in T$,
- if M is a normal form such that $u \in FTV(M)$ and $FIV(M) \cap V = \{\}$, then $M = \mathbf{throw}\ u$ L for some closed term L,
- if $M = \mathbf{catch}\ v\ L$ and $v \neq u$, then $L \in \mathcal{J}_n(u, T, V)$,
- if $M = \mathbf{throw}\ v\ L$, then $L \in \mathcal{J}_n(u, T, V)$,
- if $M = \lambda y. L$, then $L \in \mathcal{J}_n(u, T, V - \{y\})$,
- if $M = L_1\, L_2$, then $L_1, L_2 \in \mathcal{J}_n(u, T, V)$,
- if $M = <L_1, L_2>$, then $L_1, L_2 \in \mathcal{J}_n(u, T, V)$,
- if $M = \mathbf{proj}_i\ L$ for some i $(i = 1, 2)$, then $L \in \mathcal{J}_n(u, T, V)$,
- if $M = \mathbf{inj}_i\ L$ for some i $(i = 1, 2)$, then $L \in \mathcal{J}_n(u, T, V)$,
- if $M = \mathbf{case}\ L_0\ y_1.L_1\ y_2.L_2$, then
  - $L_0 \in \mathcal{J}_n(u, T, V)$, and
  - for any i $(i = 1, 2)$, there exists some V_i' such that $V_i'[FIV(L_0)/\{y_i\}] \subset V$ and $L_i \in \mathcal{J}_n(u, T, V_i')$,
- if $M = \kappa v. L$ and $v \neq u$, then $L \in \mathcal{J}_n(u, T, V)$, and
- if $M = L\, v$, then $L \in \mathcal{J}_n(u, T, V)$.

Proposition 6.2.18 *$M \in \mathcal{J}_n(u, T, V)$ if and only if*

$$\text{if } M \xrightarrow{*k} K \text{ for some } k\ (k \leq n), \text{ then } M \in \mathcal{J}_{n-k}^-(u, T, V).$$

Proof. Obvious from the definitions.

Lemma 6.2.19 *Let n be a natural number, u a tag variable, and T a type. If*

1. *$M \xrightarrow{sub*k} K$ implies $K \in \mathcal{J}_{n-k}^-(u, T, V)$,*
2. *$M \xrightarrow{sub*k} L \underset{t}{\mapsto} K$ implies $K \in \mathcal{J}_{n-k-1}(u, T, V)$, and*

3. *if* $M \overset{sub*k}{\longrightarrow} L \underset{\text{n}}{\mapsto} K$ *for some* L*, then* $K \in \mathcal{J}_{n-k-1}(u, T, V)$*,*

then $M \in \mathcal{J}_n(u, T, V)$*.*

Proof. Suppose that $M \overset{*k}{\to} K$. There are three possible cases as follows:

1. $M \overset{sub*k}{\longrightarrow} K$,
2. $M \overset{sub*l}{\longrightarrow} M'[\textbf{throw}\ v\ L/z] \underset{\text{t}}{\mapsto} \textbf{throw}\ v\ L \overset{*k-l-1}{\longrightarrow} K$ for some M', z, v and L such that $z \in FIV(M')$ and $M' \neq z$,
3. $M \overset{sub*l}{\longrightarrow} M' \underset{\text{n}}{\mapsto} L \overset{*k-l-1}{\longrightarrow} K$ for some M' and L.

It is enough to show that $K \in \mathcal{J}^-_{n-k}(u, T, V)$. In the first case, we get $K \in \mathcal{J}^-_{n-k}(u, T, V)$ from the first premise. In the second case, $K \in \mathcal{J}^-_{n-k}(u, T, V)$ because $\textbf{throw}\ v\ L \in \mathcal{J}_{n-l-1}(u, T, V)$ by the second premises. As for the third case, $L \in \mathcal{J}_{n-l-1}(u, T, V)$ from the last premise. Therefore, $K \in \mathcal{J}^-_{n-k}(u, T, V)$. □

Lemma 6.2.20 *Let* M *and* N *be terms. Let* n *be a natural number,* u *a tag variable, and* T *a type. If*

1. $U \vdash_f M\ ;\ \delta$ *is drivable,*
2. $M \in \mathcal{J}_n(u, T, V - \{x\})$*,*
3. $N \in \mathcal{X}_0$*,*
4. $N \in \mathcal{J}_n(u, T, V)$*,*
5. $x \notin \delta(u)$*, and*
6. $M[N/x] \in \mathcal{D}$*,*

then $M[N/x] \in \mathcal{J}_n(u, T, V)$*.*

Proof. By induction on the lexicographic ordering of n and $|M|$, and by cases according to the form of M. Suppose that $U \vdash_f M\ ;\ \delta$ and $N \in \mathcal{X}_0$ are drivable, $M, N \in \mathcal{J}_n(u, T, V)$, $x \notin \delta(u)$ and $M[N/x] \in \mathcal{D}$.

Case 1: $M = y$ *for some individual variable* y*.* Trivial because $M[N/x] = M$ or $M[N/x] = N$ in this case.

Case 2: $M = \textbf{catch}\ v\ M'$ *for some* v *and* M'*.* We can assume that $v \notin FTV(N) \cup \{u\}$, that is, $M[N/x] = \textbf{catch}\ v\ (M'[N/x])$. Since $U \vdash_f M\ ;\ \delta$ is derivable, $U \vdash_f M'\ ;\ \delta \sqcup \{v : V'\}$ is derivable for some V'. On the other hand, we get $M' \in \mathcal{J}_n(u, T, V - \{x\})$ from $M \in \mathcal{J}_n(u, T, V - \{x\})$, and get $x \notin (\delta \sqcup \{v : V'\})(u)$ from $x \notin \delta(u)$ because $u \neq v$. Therefore, by the induction hypothesis,

$$M'[N/x] \in \mathcal{J}_n(u, T, V). \tag{6.29}$$

We use Lemma 6.2.19 to show that $M[N/x] \in \mathcal{J}_n(u, T, V)$. For the first premise, suppose that $M[N/x] \overset{sub*k}{\longrightarrow} K = \textbf{catch}\ v\ K'$ for some K and K'. We show that $K \in \mathcal{J}^-_{n-k}(u, T, V)$. We get $K' \in \mathcal{J}_{n-k}(u, T, V)$ from (6.29). Suppose that $u \in FTV(K)$ and $FIV(K) \cap V = \{\}$. Since

$u \in FTV(K')$, $FIV(K') = FIV(K)$ and $M'[N/x] \xrightarrow{*} K'$, we get $K' = \mathbf{throw}\ u\ L$ for some closed term L from (6.29), and therefore, K is not a normal form. Thus, the first premise holds. For the second premise, suppose that $M[N/x] \xrightarrow{sub*k} L \underset{t}{\mapsto} K$ for some L. By the definition of $\xrightarrow{sub}$, we get $M'[N/x] \xrightarrow{*k+1} K$. Therefore, $K \in \mathcal{J}_{n-k-1}(u, T, V)$ from (6.29). For the last premise, suppose that $M[N/x] \xrightarrow{sub*k} L \underset{n}{\mapsto} K$ for some k ($k \leq n$) and L. There are two possible cases.

1. $\mathbf{catch}\ v\ (M'[N/x]) \xrightarrow{sub*k} \mathbf{catch}\ v\ K \underset{n}{\mapsto} K$,
2. $\mathbf{catch}\ v\ (M'[N/x]) \xrightarrow{sub*k} \mathbf{catch}\ v\ (\mathbf{throw}\ v\ K) \underset{n}{\mapsto} K$,

where $v \notin FTV(K)$ in both cases. In the first case, since $M'[N/x] \xrightarrow{sub*k+1} K$, we get $K \in \mathcal{J}_{n-k-1}(u, T, V)$ from (6.29). On the other hand, in the second case, since $M'[N/x] \xrightarrow{sub*k+1} \mathbf{throw}\ v\ K$, we also get $K \in \mathcal{J}_{n-k-1}(u, T, V)$ from (6.29). Thus, the last premise holds. We now get $M[N/x] \in \mathcal{J}_n(u, T, V)$ by Lemma 6.2.19.

Case 3: $M = \mathbf{throw}\ v\ M'$ *for some* v *and* M'. In this case, $M[N/x] = \mathbf{throw}\ v\ (M'[N/x])$. Since $U \vdash_f M\ ;\ \delta$ is derivable, $U' \vdash_f M'\ ;\ \delta'$ is derivable for some U' and δ' such that $U' \subset U$ and $\delta = \delta' \sqcup \{v : U'\}$. We can assume that $\delta = \delta'$ by Proposition 6.2.8. Note that $FIV(M') \subset U'$. On the other hand, $M' \in \mathcal{J}_n(u, T, V - \{x\})$ from $M \in \mathcal{J}_n(u, T, V - \{x\})$. Since $x \notin \delta(u) = \delta'(u)$, by the induction hypothesis,

$$M'[N/x] \in \mathcal{J}_n(u, T, V). \tag{6.30}$$

We apply Lemma 6.2.19. For the first premise, suppose that $M[N/x] \xrightarrow{sub*k} K = \mathbf{throw}\ v\ K'$ for some K and K'. We show that $K \in \mathcal{J}^-_{n-k}(u, T, V)$. We get $K' \in \mathcal{J}^-_{n-k}(u, T, V)$ from (6.30). If $u = v$, then we get $M[N/x] = M$ since $x \notin \delta(u) = \delta(v) \supset U' \supset FIV(M')$, and therefore, $K' \in T$ from $M \in \mathcal{J}_n(u, T, V - \{x\})$. Next, suppose that $u \in FTV(K)$, $FIV(K) \cap V = \{\}$ and K is a normal form. If $u \in FTV(K')$, then we get $K' = \mathbf{throw}\ u\ L$ for some L from (6.30). Since K is normal, we get $u \notin FTV(K')$, that is, $u = v$ and $M[N/x] = M$, again. This implies that K' is a closed term since $M \in \mathcal{J}_n(u, T, V - \{x\})$. For the second premise, suppose that $M[N/x] \xrightarrow{sub*k} L \underset{t}{\mapsto} K$ for some L. By the definition of $\xrightarrow{sub}$, we get $M'[N/x] \xrightarrow{*k+1} K$, and therefore, $K \in \mathcal{J}_{n-k-1}(u, T, V)$ from (6.30). For the last premise, it is impossible that $M[N/x] \xrightarrow{sub*k} L \underset{n}{\mapsto} K$ by the definition of $\xrightarrow{sub}$ and $\underset{n}{\mapsto}$. Therefore, we get $M[N/x] \in \mathcal{J}_n(u, T, V)$ by Lemma 6.2.19.

Case 4: $M = \lambda y.\, M'$ *for some* y *and* M'. We can assume that $y \notin FIV(N)$, that is, $M[N/x] = \lambda y.\, (M'[N/x])$. Since $U \vdash_f M\ ;\ \delta$ is derivable, so is $U \cup \{y\} \vdash_f M'\ ;\ \delta$. We get $M' \in \mathcal{J}_n(u, T, V - \{x\} - \{y\})$ from $M \in \mathcal{J}_n(u, T, V - \{x\})$. On the other hand, $N \in \mathcal{J}_n(u, T, V - \{y\})$ since $y \notin FIV(N)$. Therefore, by the induction hypothesis,

$$M'[N/x] \in \mathcal{J}_n(u, T, V - \{y\}). \tag{6.31}$$

We use Lemma 6.2.19 to show $\lambda y.\, (M'[N/x]) \in \mathcal{J}_n(u, T, V)$. For the first premise, suppose that $M[N/x] \xrightarrow{sub*k} K$. By the definition of $\xrightarrow{sub}$, we get $K = \lambda y.\, K'$ for some K' such that $M'[N/x] \xrightarrow{*k} K'$. We get $K' \in \mathcal{J}_{n-k}(u, T, V - \{y\})$ from (6.31). On the other hand, suppose that $u \in FTV(K)$, $FIV(K) \cap V = \{\}$ and K is a normal form, that is, $u \in FTV(K')$, $FIV(K') \cap (V - \{y\}) = \{\}$ and K' is normal. Therefore, $K' = \mathbf{throw}\ u\ L$ for some closed term L from (6.31), and this implies $K \underset{t}{\mapsto} K'$, that is, a contradiction. Thus, the first premise holds. For the second premise, suppose that $M[N/x] \xrightarrow{sub*k} L \underset{t}{\mapsto} K$ for some L, that is, $M'[N/x] \xrightarrow{*k+1} K$. Therefore,

$K \in \mathcal{J}_{n-k-1}(u, T, V - \{y\})$ from (6.31). We get $K \in \mathcal{J}_{n-k-1}(u, T, V)$ by Proposition 6.1.11. Since the last premise is trivial, we get $M[N/x] \in \mathcal{J}_n(u, T, V)$ by Lemma 6.2.19.

Case 5: *$M = M_1\, M_2$ for some M_1 and M_2.* In this case, $M[N/x] = (M_1[N/x])\,(M_2[N/x])$. Since $U \vdash_f M\,;\,\delta$ is derivable, $U_1 \vdash_f M_1\,;\,\delta_1$ and $U_2 \vdash_f M_2\,;\,\delta_2$ are derivable for some U_1, U_2, δ_1 and δ_2 such that $U = U_1 \cup U_2$ and $\delta = \delta_1 \sqcup \delta_2$. We also get $M_1, M_2 \in \mathcal{J}_n(u, T, V - \{x\})$ from $M \in \mathcal{J}_n(u, T, V - \{x\})$. Therefore, by the induction hypothesis,

$$M_1[N/x],\ M_2[N/x] \in \mathcal{J}_n(u, T, V). \tag{6.32}$$

We use Lemma 6.2.19. For the first premise, suppose that $M[N/x] \xrightarrow{sub*k} K = K_1\, K_2$, that is, $M_i[N/x] \xrightarrow{*k} K_i$ for any i ($i = 1, 2$). We get $K_1, K_2 \in \mathcal{J}_{n-k}(u, T, V)$ from (6.32). On the other hand, suppose that $u \in FTV(K)$, $FIV(K) \cap V = \{\}$ and K is a normal form, that is, $u \in FTV(K_i)$, $FIV(K_i) \cap V = \{\}$ and K_i is normal for some i. Therefore, $K_i =$ **throw** u L for some closed term L from (6.32), and this implies $K \underset{\mathrm{t}}{\mapsto} K_i$, that is, a contradiction. For the second premise, suppose that $M[N/x] \xrightarrow{sub*k} L \underset{\mathrm{t}}{\mapsto} K$ for some L, that is, $M_i[N/x] \xrightarrow{*k+1} K$ for some i ($i = 1, 2$). We get $K \in \mathcal{J}_{n-k-1}(u, T, V)$ from (6.32). For the last premise, suppose that $M[N/x] \xrightarrow{sub*k} L \underset{\mathrm{n}}{\mapsto} K$, that is, for some y, L_1 and L_2,

$$(M_1[N/x])\,(M_2[N/x]) \xrightarrow{sub*k} (\lambda\, y.\, L_1)\, L_2 \underset{\mathrm{n}}{\mapsto} L_1[L_2/y] = K.$$

By Proposition 6.1.10, we get $\lambda\, y.\, L_1 \in \mathcal{J}_{n-k}(u, T, V)$ and $L_2 \in \mathcal{J}_{n-k}(u, T, V)$ from (6.32). Therefore, $L_1 \in \mathcal{J}_{n-k}(u, T, V - \{y\})$ and $L_2 \in \mathcal{J}_{n-k}(u, T, V)$. By Proposition 6.1.10 and Proposition 6.1.11,

$$L_1,\ L_2 \in \mathcal{J}_{n-k-1}(u, T, V). \tag{6.33}$$

On the other hand, $U' \vdash_f M_1[N/x]\,;\,\delta'$ is derivable for some U' and δ' by Proposition 6.2.10 since $M_1, N \in \mathcal{X}_0$, and therefore $U' \vdash_f \lambda\, y.\, L_1\,;\,\delta'$ is also derivable by Proposition 6.2.14 since $M_1[N/x] \xrightarrow{*} \lambda\, y.\, L_1$. That is,

$$U' \cup \{y\} \vdash_f L_1\,;\,\delta' \tag{6.34}$$

is derivable and

$$y \notin \delta'(u). \tag{6.35}$$

As for L_2, we get $M_2[N/x] \in \mathcal{X}_0$ from $M_2, N \in \mathcal{X}_0$ by Proposition 6.2.10. That is, $L_2 \in \mathcal{X}_0$ by Proposition 6.2.14. Moreover, we get $L_1[L_2/y] \in \mathcal{D}$ from $M[N/x] \in \mathcal{D}$ since $M[N/x] \xrightarrow{*} L_1[L_2/y]$. Therefore, by the induction hypothesis, we get

$$K = L_1[L_2/y] \in \mathcal{J}_{n-k-1}(u, T, V)$$

from (6.33), (6.34) and (6.35). Thus, the last premise also holds. We now get $M[N/x] \in \mathcal{J}_n(u, T, V)$ by Lemma 6.2.19.

Case 6: *$M = \kappa\, v.\, M'$ for some v and M'.* We can assume that $v \notin FTV(N) \cup \{u\}$, that is, $M[N/x] = \kappa\, v.\, (M'[N/x])$. Since $U \vdash_f M\,;\,\delta$ is derivable, so is $U \vdash_f M'\,;\,\delta \sqcup \{v\!:\!V'\}$ for some V'. On the other hand, we get $M' \in \mathcal{J}_n(u, T, V - \{x\})$ from $M \in \mathcal{J}_n(u, T, V - \{x\})$, and get $x \notin (\delta \sqcup \{v\!:\!V'\})(u)$ from $x \notin \delta(u)$ because $u \neq v$. Therefore, by the induction hypothesis,

$$M'[N/x] \in \mathcal{J}_n(u, T, V). \tag{6.36}$$

We use Lemma 6.2.19 For the first premise, suppose that $M[N/x] \xrightarrow{sub*k} K = \kappa v.\, K'$. We show that $K \in \mathcal{J}^-_{n-k}(u, T, V)$. We get $K' \in \mathcal{J}^-_{n-k}(u, T, V)$ from (6.36) since $M'[N/x] \xrightarrow{*k} K'$. If $u \in FTV(K)$ and $FIV(K) \cap V = \{\}$, then since $u \in FTV(K')$, we get $K' = \textbf{throw}\ u\ L$ for some L from (6.36), that is, K is not normal. Therefore, the first premise holds. For the second premise, suppose that $M[N/x] \xrightarrow{sub*k} L \underset{t}{\mapsto} K$ for some L. By the definition of $\xrightarrow{sub}$, we get $M'[N/x] \xrightarrow{*k+1} K$, and therefore, $K \in \mathcal{J}_{n-k-1}(u, T, V)$ from (6.30). Since the last premise is trivial, we get (6.2) by Lemma 6.2.19.

Case 7: *$M = M'\, v$ for some M' and v.* In this case, $M[N/x] = (M'[N/x])\, v$. Since $U \vdash_f M\, ;\, \delta$ is derivable, $U' \vdash_f M'\, ;\, \delta'$ is derivable for some U' and δ' such that $U' \subset U$ and $\delta = \delta' \sqcup \{v : U'\}$. We can assume that $\delta = \delta'$ by Proposition 6.2.8. Note that $FIV(M') \subset U'$. On the other hand, $M' \in \mathcal{J}_n(u, T, V - \{x\})$ from $M \in \mathcal{J}_n(u, T, V - \{x\})$. Since $x \notin \delta(u) = \delta'(u)$, by the induction hypothesis,

$$M'[N/x] \in \mathcal{J}_n(u, T, V). \tag{6.37}$$

We use Lemma 6.2.19 For the first premise, suppose that $M[N/x] \xrightarrow{sub*k} K = K'\, v$. We get $K' \in \mathcal{J}^-_{n-k}(u, T, V)$ from (6.37) since $M'[N/x] \xrightarrow{*k} K'$. Suppose that $u \in FTV(K)$ and $FIV(K) \cap V = \{\}$. We show that K is not normal. If $u \in FTV(K')$, then $K = \textbf{throw}\ u\ L$ for some closed term L by (6.37). On the other hand, $u \notin FTV(K')$ implies $u = v$, and therefore, $M[N/x] = M$ since $x \notin \delta(u) = \delta(v) \supset U' \supset FIV(M')$. Therefore, K is not normal since $M \in \mathcal{J}_n(u, T, V - \{x\})$ and $M \xrightarrow{*k} K = K'\, v$. For the second premise, suppose that $M[N/x] \xrightarrow{sub*k} L \underset{t}{\mapsto} K$ for some L. By the definition of $\xrightarrow{sub}$, we get $M'[N/x] \xrightarrow{*k+1} K$, and therefore, $K \in \mathcal{J}_{n-k-1}(u, T, V)$ from (6.37). For the last premise, suppose that $M[N/x] \xrightarrow{sub*k} L \underset{n}{\mapsto} K$, that is, for some L' and w,

$$M'[N/x]\, v \xrightarrow{sub*k} (\kappa w.\, L')\, v \underset{n}{\mapsto} L'[v/w] = K. \tag{6.38}$$

We can assume that $w \neq u$. Since $M'[N/x] \xrightarrow{*k} \kappa w.\, L'$, we get $\kappa w.\, L' \in \mathcal{J}_{n-k}(u, T, V)$ from (6.37), that is,

$$L' \in \mathcal{J}_{n-k}(u, T, V).$$

Therefore, if $v \neq u$, then we get $K = L'[w/v] \in \mathcal{J}_{n-k}(u, T, V)$ by Proposition 6.1.14, that is, $K \in \mathcal{J}_{n-k-1}(u, T, V)$ by Proposition 6.1.10. On the other hand, if $v = u$, then we get $M[N/x] = M$ since $x \notin \delta(u) = \delta(v) \supset U' \supset FIV(M')$. Therefore, $K \in \mathcal{J}_{n-k-1}(u, T, V)$ since $M \in \mathcal{J}_n(u, T, V - \{x\})$ and $M \xrightarrow{*k+1} K$. Thus, the last premise holds. We now get $M[N/x] \in \mathcal{J}_n(u, T, V)$ by Lemma 6.2.19.

Case 8: *M has one of the other forms.* The proof is similar. □

Theorem 6.2.21 *X_0 is an admissible frame.*

Proof. Since X_0 is a frame by Proposition 6.2.16, we show that $\mathcal{X}_0$ is admissible. Suppose that

$$M,\, N \in \mathcal{X}_0 \cap \mathcal{J}_n(u, T, V), \quad \text{and} \tag{6.39}$$

$$M\, N \in \mathcal{D}. \tag{6.40}$$

By Proposition 6.1.17, it is enough to show that

$$M\, N \in \mathcal{J}_n(u, T, V).$$

We use Lemma 6.2.19. For the first premise, suppose that $M\,N \overset{sub*k}{\longrightarrow} K = K_1\,K_2$, that is, $M \overset{*k}{\to} K_1$ and $N \overset{*k}{\to} K_2$. We get $K_1, K_2 \in \mathcal{J}_{n-k}(u, T, V)$ from (6.39). On the other hand, suppose that $u \in FTV(K)$, $FIV(K) \cap V = \{\}$ and K is a normal form. Since $u \in FTV(M)$ and $FIV(K_1) \cap V = \{\}$, or $u \in FTV(N)$ and $FIV(K_2) \cap V = \{\}$, we get $M = \textbf{throw}\ u\ L$ or $N = \textbf{throw}\ u\ L$ for some closed term L from (6.39), and this implies $K \underset{\mathrm{t}}{\mapsto} \textbf{throw}\ u\ L$, that is, a contradiction. For the second premise, suppose that $M\,N \overset{sub*k}{\longrightarrow} L \underset{\mathrm{t}}{\mapsto} K$ for some L, that is, $M \overset{*k+1}{\longrightarrow} K$ or $N \overset{*k+1}{\longrightarrow} K$. We get $K \in \mathcal{J}_{n-k-1}(u, T, V)$ from (6.39). For the last premise, suppose that $M\,N \overset{sub*k}{\longrightarrow} L \underset{\mathrm{n}}{\mapsto} K$, that is, for some y, L_1 and L_2,

$$M\,N \overset{sub*k}{\longrightarrow} (\lambda\,y.\,L_1)\,L_2 \underset{\mathrm{n}}{\mapsto} L_1[L_2/y] = K.$$

We get $\lambda\,y.\,L_1 \in \mathcal{J}_{n-k}(u, T, V)$ and $L_2 \in \mathcal{J}_{n-k}(u, T, V)$ from (6.39) by Proposition 6.1.10. Therefore, $L_1 \in \mathcal{J}_{n-k}(u, T, V - \{y\})$ and $L_2 \in \mathcal{J}_{n-k}(u, T, V)$. By Proposition 6.1.10,

$$L_1 \in \mathcal{J}_{n-k-1}(u, T, V - \{y\}), \tag{6.41}$$

$$L_2 \in \mathcal{J}_{n-k-1}(u, T, V). \tag{6.42}$$

On the other hand, $\lambda\,y.\,L_1 \in \mathcal{X}_0$ by Proposition 6.2.14 since $M \overset{*}{\to} \lambda\,y.\,L_1$. That is, for some U and δ,

$$U \cup \{y\} \vdash_f L_1\,;\,\delta \tag{6.43}$$

is derivable and

$$y \notin \delta(u). \tag{6.44}$$

As for L_2, we get $L_2 \in \mathcal{X}_0$ by Proposition 6.2.14 since $N \in \mathcal{X}_0$. Moreover, we get $L_1[L_2/y] \in \mathcal{D}$ from $M\,N \in \mathcal{D}$ since $M\,N \overset{*}{\to} L_1[L_2/y]$. Therefore, by Lemma 6.2.20, we get

$$L_1[L_2/y] \in \mathcal{J}_{n-k-1}(u, T, V)$$

from (6.41), (6.42), (6.43) and (6.44). Thus, the last premise also holds. Therefore, we get $M\,N \in \mathcal{J}_n(u, T, V)$ by Lemma 6.2.19. □

6.3 Strong normalizability and normal forms

We get strong normalizability of well-typed terms from the discussion on the term model. The term model also provides a results on the form of well-typed normal forms.

Theorem 6.3.1 *Let M be a term. If $\Gamma \vdash M : C\,;\,\Delta$ is derivable for some Γ, C and Δ, then M is strongly normalizable.*

Proof. Since $x \in \mathcal{I}(\mathcal{X}_0, \Gamma(x))$ for any $x \in Dom(\Gamma)$, $M \in \mathcal{I}(\mathcal{X}_0, C)$ by Theorem 6.1.33 on $\mathcal{X}_0$. Therefore, M is strongly normalizable by the definition of $\mathcal{I}(\mathcal{X}_0, C)$. □

Theorem 6.3.2 *Let M be a normal form such that $FIV(M) = \{\}$. If $\Gamma \vdash M : C\,;\,\Delta$ is derivable for some Γ, C and Δ, then one of the following holds.*

1. *$M = \textbf{throw}\ u\ L$ for some closed term L.*

2. *C is an atomic formula, and M is a constant.*

3. *$C = A \supset B$ for some A and B, and $M = \lambda y.\, L$ for some y and L.*

4. *$C = A \wedge B$ for some A and B, and $M = <L_1,\ L_2>$ for some L_1 and L_2.*

5. *$C = A \vee B$ for some A and B, and $M = \mathbf{inj}_i\ L$ for some i ($i = 1, 2$) and L.*

6. *$C = A \triangleleft B$ for some A and B, and $M = \kappa\, v.\, L$ for some v and L.*

Proof. Let $x_1, \ldots, x_m$ be as $Dom(\Gamma) = \{x_1, \ldots, x_2\}$, and suppose that $\Gamma \vdash M : C\ ;\ \Delta$ is derivable. Since $FIV(M) = \{\}$, $\{\} \vdash M : C\ ;\ \Delta[\{\}/\{x_1\}, \ldots, \{\}/\{x_m\}]$ is also derivable by Proposition 5.4.14. Therefore, by Theorem 6.1.33 on $\mathcal{X}_0$,

$$M \in \mathcal{I}(\mathcal{X}_0,\ C) \cap \mathcal{J}(u,\ \mathcal{I}(\mathcal{X}_0,\ \Delta^t(u)),\ \{\})$$

for any u. Since M is normal and $FIV(M) = \{\}$, if $u \in FTV(M)$, then $M = \mathbf{throw}\ u\ L$ for some closed term L. On the other hand, if $FTV(M) = \{\}$, then one of 2 through 6 holds since $M \in \mathcal{I}^-(\mathcal{X}_0,\ C)$ and $FIV(M) = \{\}$. □

6.4 Realizability interpretation of $L_{c/t}$

Let $\mathcal{X}$ be an admissible frame, and let $\mathcal{A}$ be a mapping which assigns a subset of *Const* to each atomic type. The realizability for $L_{c/t}$ is defined relatively to $\mathcal{X}$ and $\mathcal{A}$.

Definition 6.4.1 (Realizability of types) Let M be a term, and A a type. We define a relation $\mathbf{r}$ between terms and types as follows.

$$M\ \mathbf{r}\ A \quad \text{iff} \quad M \in \mathcal{I}(\mathcal{X},\ A).$$

Definition 6.4.2 (Realizability of tag contexts) Let M be a term, and Δ a tag context. We define a relation $\mathbf{r}$ between terms and tag contexts as follows.

$$M\ \mathbf{r}\ \Delta \quad \text{iff} \quad \text{for any } u,\ M \in \mathcal{J}(u,\ \mathcal{I}(\mathcal{X},\ \Delta^t(u)),\ \Delta^v(u)),$$

where $\mathcal{I}(\mathcal{X},\ \Delta^t(u)) = \{\}$ and $\Delta^v(u) = \{\}$ if $u \notin Dom(\Delta)$.

Definition 6.4.3 (Interpretation) We define the interpretation of typing judgements as follows. The relation

$$\{x_1 : A_1, \ldots, x_m : A_m\} \models M : C\ ;\ \{u_1 : B_1^{V_1}, \ldots, u_n : B_n^{V_n}\}$$

holds if and only if for any terms $K_1, \ldots, K_m$ such that $K_i\ \mathbf{r}\ A_i$ and $x_j \notin FIV(K_i)$ for any i and j $(1 \le i, j \le m)$,

1. $M[K_1/x_1, \ldots, K_m/x_m]\ \mathbf{r}\ C$, and
2. if $K_i\ \mathbf{r}\ \Delta$ for any i $(1 \le i \le m)$, then

$$M[K_1/x_1, \ldots, K_m/x_m]\ \mathbf{r}\ \Delta[FIV(K_1)/\{x_1\}, \ldots, FIV(K_m)/\{x_m\}].$$

The following soundness theorem assures us that we can regard the derivations of $L_{c/t}$ as programs which satisfy the specification defined by the realizability interpretation defined above.

Theorem 6.4.4 (Soundness of $L_{c/t}$) *If $\Gamma \vdash M : C\,;\Delta$ is derivable in $L_{c/t}$, then $\Gamma \models M : C\,;\Delta$ holds.*

Proof. Straight forward from Theorem 6.1.33. □

Chapter 7

Concluding remarks

We have presented two typing systems $L_{c/t}^{\text{CBV}}$ and $L_{c/t}$ which capture the catch/throw mechanism in the notion of "proofs as programs". Although they are just variants of the standard constructive logic, they admit extra conclusions besides the main one. By this feature, we can naturally construct proofs which handle the exceptional situations efficiently as in practical programming languages. We showed the direct correspondence between such proofs and programs which make use of the catch and throw mechanism by certain realizability interpretations. The soundness theorems of the systems relative to these interpretations assure that they can still be basises for the formal method of computer programming. Moreover, the non-determinism introduced with the catch and throw mechanism does not break this paradigm, because we can take the meanings of programs by a realizability interpretation independent of the evaluation strategy, that is, reductions do not preserve the meaning of programs as values, but do preserve the meaning as realizers.

From a computational point of view, the catch and throw mechanism provides only a restricted access to the current continuation, and does not provide the full access as the first class objects. Therefore, it could be regarded as a trivial subcase of more powerful facilities such as *call/cc* of Scheme. However, from the viewpoint of the notion of "proofs as programs", it assures correct programs without any of the restrictions required for the case of such more powerful ones [11, 20, 21]. And more importantly, the catch and throw mechanism has a natural counterpart in the reasoning of programmers, that is, a characteristic way of exception handling. We doubt whether there also exists such a natural reasoning corresponding to the use of *call/cc* and its variants beyond the catch/throw.

From the viewpoint of program verification, our work can be regarded as a higher-order extension of the work concerning `goto` statements in Hoare's logic, whose main idea is also the existence of extra post-conditions [1, 2]. However, it should be noted that our work captures the logic of programmers behind the facilities for non-local exit rather than their computational behavior. In this sense, the catch and throw mechanism is just a sample of possible realizers for our logic.

There remain some problems which have not been discussed in this thesis. We have considered only a propositional fragment, and have not investigated the relation between the catch and throw mechanism and mathematical inductions. Actually, the mechanism is often used in subroutines that call themselves recursively in practical programming. We could expect that the standard rules for mathematical inductions would work as well from just a logical point of view, but more intensive investigation of the practice should be done in order to capture the class of proofs, that is, the class of programs, used in practical programming.

Bibliography

[1] S. Alagić and M. A. Arbib, *The Design of Well-Structured and Correct Programs*, Springer-Verlag, 1978.

[2] M. A. Arbib and S. Alagić, Proof Rules for Gotos, *Acta Informatica*, Vol. 11, pp. 139-148, 1979.

[3] F. Barbanera ad S. Berardi, A Symmetric Lambda Calculus for "Classical" Program Extraction, *Theoretical Aspects of Computer Software*, M. Hagiya and J. C. Mitchell, eds., Lecture Notes in Computer Science 789, pp. 495-515, Springer-Verlag, 1994.

[4] H. P. Barendregt, *The Lambda Calculus: Its Syntax and Semantics*, North-Holland, 1984.

[5] R. L. Constable, et al., *Implementing Mathematics with the Nuprl Proof Development System*, Prentice-Hall, 1986.

[6] T. Coquand and G. Huet, The Calculus of Construction, *Information and Computation*, Vol. 76, pp. 95-120, 1988.

[7] M. Felleisen, D. Friedman, E. Kohlbecker, and B. Duba, A syntactic theory of sequential control, *Theoretical Computer Science*, Vol. 52(3), pp. 205-237, 1987.

[8] H. Friedman, Classically and intuitionistically provably recursive functions, *Higher Set Theory*, D. S. Scott and G. H. Muller, eds., Lecture Notes in Mathematics 699, pp. 21-28, Springer-Verlag, 1978.

[9] J.-Y. Girard, A new constructive logic: classical logic, *Mathematical Structures in Computer Science*, Vol. 1, pp. 255-296, 1991.

[10] J.-Y. Girard, Y. Lafont and P. Taylor, *Proofs and Types*, Cambridge University Press, 1989.

[11] T. G. Griffin, A formulae-as-types notion of control, *Conf. Rec. ACM Symp. on Principles of Programming Languages*, pp. 47-58, 1990.

[12] S. Hayashi, Singleton, Union and Intersection Types for Programs, *Theoretical Aspects of Computer Software*, T. Ito and A.R. Meyer, eds., Lecture Notes in Computer Science 526, pp. 701-730, Springer-Verlag, 1991.

[13] S. Hayashi and H. Nakano, *PX: A Computational Logic*, The MIT Press, 1988.

[14] W. A. Howard, The Formulae-as-types Notion of Constructions, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pp. 479-490, Academic Press, 1980.

[15] B. W. Kernighan and D. M. Ritchie, *The C programming language (2nd ed.)*, Prentice-Hall, 1989.

[16] S. C. Kleene, *Introduction to Metamathematics*, North-Holland, 1952.

[17] P. J. Landin, The mechanical evaluation of expressions, *Computer Journal*, Vol. 6(4), 1964.

[18] S. Maehara, Eine Darstellung intuitionistischen Logik in der Klassishen, *Nagoya Math. Journal*, Vol. 7, pp. 45-64, 1954.

[19] P. Martin-Löf, Constructive mathematics and computer programming, *Logic, Methodology, and Philosophy of Science VI*, L. J. Choen, et al., eds., North-Holland, pp. 153-179, 1982.

[20] C. R. Murthy, An evaluation semantics for classical proofs, *Proc. 6th Annual IEEE Symp. on Logic in Computer Science*, pp. 96-107, 1991.

[21] C. R. Murthy, Classical Proofs as Programs: How, What and Why, *Technical Report TR 91-1215*, Department of Computer Science, Cornell University, 1991.

[22] C. R. Murthy, A Computational Analysis of Girard's Translation and LC, *Proc. 7th Annual IEEE Symp. on Logic in Computer Science*, pp. 72-81, 1992.

[23] H. Nakano, A Constructive Formalization of the Catch and Throw Mechanism, *Proc. 7th Annual IEEE Symp. on Logic in Computer Science*, pp. 82-89, 1992.

[24] H. Nakano, The Non-deterministic Catch and Throw Mechanism and Its Subject Reduction Property, *Logic, Language and Computation*, N. D. Jones, et al., eds., Lecture Notes in Computer Science 792, pp. 61-72, Springer-Verlag, 1994.

[25] H. Nakano, A constructive logic behind the the catch and throw mechanism, *Annals of Pure and Applied Logic*, Vol. 69, pp. 269-301, 1994.

[26] N. N. Nepejvoda, A bridge between constructive logic and computer programming, *Theoretical Computer Science*, Vol. 90, pp. 253-270, 1991.

[27] M. Parigot, Free Deduction: An Analysis of "Computations" in Classical Logic, *Proc. Russian Conf. on Logic Programming*, A. Voronkov, ed., Lecture Notes in Computer Science 592, pp. 361-380, Springer-Verlag, 1991.

[28] M. Parigot, $\lambda\mu$-Calculus: An Algorithmic Interpretation of Classical Natural Deduction, *Proc. Int'l Conf. on Logic Programming and Automated Reasoning*, Lecture Notes in Computer Science 624, pp. 190-201, Springer-Verlag, 1992.

[29] M. Parigot, Classical Proof as Programs, *Computational logic and theory*, Lecture Notes in Computer Science 713, G. Gottlob, et al., eds., pp. 263-276, Springer-Verlag, 1993.

[30] G. D. Plotkin, Call-by-name, call-by-value and the λ-calculus, *Theoretical Computer Science*, Vol. 1, pp. 125-159, 1975.

[31] N. J. Rehof and M. H. Sørensen, The λ_Δ-calculus, *Theoretical Aspects of Computer Software*, M. Hagiya and J. C. Mitchell, eds., Lecture Notes in Computer Science 789, pp. 516-542, Springer-Verlag, 1994.

[32] G. L. Steele, *Common Lisp: The Language*, Digital Press, 1984.

[33] K. Schütte, *Vollständige Systeme Modaler und Intuitionistischer Logik*, Springer-Verlag, 1968.

[34] G. Takeuti, *Proof theory (2nd ed.)*, North Holland, 1987.

[35] Y. Takayama, Extraction of Redundancy-free Programs from Constructive Natural Deduction Proofs, *Journal of Symbolic Computation*, Vol. 12, pp. 29-69, 1991.

[36] M. Tatsuta, Program Synthesis Using Realizability, *Theoretical Computer Science*, Vol. 90, pp. 309-353, 1991.