AN APPLICATION OF NDJ_{PROP} TO CATCH AND THROW MECHANISM

MASAHIRO NAKATA
Kobe University
1-1 Rokkodai, Nada, Kobe, 657-8501 Japan
E-mail: cavs@mint.freemail.ne.jp

NORIMICHI SANETO
Setoda Highschool
Setodachou, Toyotagun, Hiroshima 722-2417 Japan
E-mail: saneto@lime.ocn.ne.jp

MARIKO YASUGI
Kyoto Sangyo University
Motoyama, Kamigamo, Kita, Kyoto 603-8555 Japan
E-mail: yasugi@cc.kyoto-su.ac.jp

"Catch and throw" is a mechanism in the program language dealing with exceptional processes. Nakano in [4, 5, 6, 7] introduced a typing system, named $L_{c/t}^{CBV}$, which can express the exceptional processes. He proved its soundness by interpreting it with a term (program) system with the catch and throw operators. We will here take the advantage of a system of the propositional calculus, named NDJ_{prop}, and extend it to obtain a deductive system $\mathrm{NDJ}_{c/t}$ as an alternative to $L_{c/t}^{CBV}$. We show that there are mutual maps between Nakano's term system and $\mathrm{NDJ}_{c/t}$ which preserve reductions (in a certain sense), thus establishing the "proofs-as-programs" paradigm directly.

1. Introduction

Nakano in [4, 5, 6, 7] defined a term system (locally named $\mathcal{T}$ here) which contains the constructs of "catch" and "throw", and also he introduced a "typing system", named $L_{c/t}^{CBV}$, which can represent the "catch and throw" mechanism. With these, he first gave an operational semantics of $\mathcal{T}$ and next proved the soundness of $L_{c/t}^{CBV}$ by a realization interpretation in terms of $\mathcal{T}$. He also reformulated $L_{c/t}^{CBV}$ in the sequential style, proving its cut-elimination theorem.

The contribution of the present article is to formulated a formal system $\mathrm{NDJ}_{c/t}$ that corresponds to the term system $\mathcal{T}$ in the natural deduction style, and define maps between the terms of $\mathcal{T}$ and the proofs of $\mathrm{NDJ}_{c/t}$ that preserve reductions (in a certain sense). This way, the "proofs-as-programs" paradigm is established directly. The system $\mathrm{NDJ}_{c/t}$ is an extension of NDJ_{prop}, the constructive propositional calculus formulated in the natural deduction style.

The map from $\mathrm{NDJ}_{c/t}$ to $\mathcal{T}$ mentioned above is expected to be useful also for the "proof animation", proposed by Hayashi in [2], for example. Namely, the validity of a formal (supposed) proof is to be tested by computing the term (program) extracted from the (supposed) proof for some examples of inputs. The map from $\mathcal{T}$ to $\mathrm{NDJ}_{c/t}$ will be also useful in order to give a proper proof once a program is (if erroneous) corrected.

In [13, 14, 15], Yasugi, Ryu and Nakata proposed and investigated a system of the classical predicate calculus in the natural deduction style, named NDK, whose characteristic is that introductions and eliminations of any logical connective can be executed at any place in the disjunctive components of a formula. This way, the law of excluded middle can be deduced as the consequence of an inference, not as an axiom. For example, the inference rule $\wedge$-elimination applies to $C \vee (A \wedge B) \vee D$ to deduce the consequence $C \vee A \vee D$; the inference rule $\neg$-introduction applies to $B \vee C$ with a live assumption $[A]$ to deduce the consquence $B \vee \neg A \vee C$ with $[A]$ discharged.

In [8] Nakata considered the propositional part of NDK, named NDK_{prop}, and proposed a variation of the implicational part of NDK_{prop}, named $\mathrm{NDK}^{\rightarrow}$, which suited better to deal with reductions. In [11], Saneto made some observations of the constructive part of NDK_{prop}, named NDJ_{prop}, which is obtained from NDK_{prop} by restricting the introduction of $\rightarrow$ or $\neg$ applied only to the whole formula in the premise. He then applied it to define a system $\mathrm{NDJ}_{c/t}$, which expresses the "catch and throw" mechanism, modelling after Nakano's method in [4, 5, 6, 7]. With $\mathrm{NDJ}_{c/t}$, the correspondences between a term system and a logical system immediately follow. The map from the deductive system $\mathrm{NDJ}_{c/t}$ to the term system $\mathcal{T}$ is modelled after Nakano's idea, while the converse map is modelled after [12].

This article has emerged from Sections 6 and 7 of [11].

In Section 2 of this article, we introduce the system $\mathrm{NDJ}_{c/t}$; in Section 3, we first present the definition of the term system $\mathcal{T}$, and then define correspondences between the terms of $\mathcal{T}$ and the proofs of $\mathrm{NDJ}_{c/t}$ (Propo-

sitions 1 and 2, Theorem). At the end, in Section 4, Nakano's example of realizing the "catch and throw" is presented in our language.

A related work is seen in [3]. A comprehensive report on the propositional subsystems of NDK has been given in [9]. We have consulted [1] and [10] for systems of natural deductions.

2. A system $\mathrm{NDJ}_{c/t}$ representing catch and throw

In the follwing, we will define the system $\mathrm{NDJ}_{c/t}$. $\mathrm{NDJ}_{c/t}$is an expansion of NDJ_{prop}, a formulation of the constructive propositional logic, to a system which can express the catch and throw mechanism. In the conclusion of an inference of $\mathrm{NDJ}_{c/t}$, there occurs a formula of the form $A \vee \Delta^T$. Here T represents a set of assignments of *tag* variables. Tag variables, a new logical connective $\triangleleft$ and new rules of inferences are added to NDJ_{prop}, based on Nakano's ideas ([4, 5, 6, 7]).

Definition 2.1.

1) The propositional formulas are defined as usual with connectives $\wedge, \vee, \rightarrow$.
2) A new logical connective $\triangleleft$ will be introduced, so that, if A and B are formulas, then $A \triangleleft B$ is also a formula. It is called "exception" by Nakano.
3) *Tag variables* are denoted by $u, v, \cdots$. A formula B with an assignment of a tag variable u is denoted as B^u. B and B^u are identical as propositions, but their roles are different. A formula with a tag assigned is called an *error message*.
4) The disjunctive components of a formula A are defined as follows.
   1. If A is an atomic formula, then the disjunctive component of A is A itself.
   2. If A is of the form $B \vee C$, then disjunctive components of A are disjunctive components of B, disjunctive components of C and A itself.
   3. If A is of the form $B \wedge C$, $B \rightarrow C$ or $B \triangleleft C$, then the disjunctive component of A is A itself. (Examples of disjunctive components are listed below.)
5) The formal proof of $\mathrm{NDJ}_{c/t}$ will be defined below. Labels of assumptions will be denoted by $x, y, \cdots$. The signs such as $(x), (x, y)$ writen at the left of the name of an inference rule will indicate the label(s) of discharged assumption(s).

Let us note that, in an assumption of an NDJ$_{c/t}$-proof as defined below, no error message occurs. In the conclusion of an inference, a formula of the form $A \vee (B_1^{u_1} \vee B_2^{u_2} \vee \cdots \vee B_n^{u_n})$ may occur. Here, $n \geq 0$, and we may abbreviate a disjunctive formula with tags like $(B_1^{u_1} \vee B_2^{u_2} \vee \cdots \vee B_n^{u_n})$ as Δ^T. In $A \vee \Delta^T$, A will be called the main part, and Δ^T will be called the exceptional part.

Proofs of NDJ$_{c/t}$: A proof of NDJ$_{c/t}$ consists of "assumptions" and "inference rules" as listed below.

[Assumption] $[x : A]$ will denote an "assumption" A (a formula without tag variables) with the label x.

[Inference rules of NDJ$_{c/t}$]

$\Delta^T(E^u)$ will represent a formula with E^u added to Δ^T as a "disjunctive component," while $\Delta^T(\widehat{E^u})$ represents simply Δ^T but indicates the place where E^u is added. For example, if $\Delta^T(E^u)$ is $X^u \vee E^u \vee Y^u$, then $\Delta^T(\widehat{E^u})$ is $X^u \vee Y^u$.

$$\frac{\vdots}{\dfrac{A \vee \Delta^T(\widehat{E^u})}{A \vee \Delta^T(E^u)}\ weak} \qquad \frac{\begin{array}{c}[x:A]\\ \vdots \quad \vdots\\ A \vee \Delta^T \quad C \vee \Delta^T\end{array}}{C \vee \Delta^T}\ (x) - let$$

$$\frac{\vdots}{\dfrac{E \vee \Delta^T(\widehat{E^u})}{A \vee \Delta^T(E^u)}\ throw} \qquad \frac{\vdots}{\dfrac{A \vee \Delta^T(A^u)}{A \vee \Delta^T(\widehat{A^u})}\ catch}$$

$$\frac{\begin{array}{cc}\vdots & \vdots\\ A \vee \Delta^T & B \vee \Delta^T\end{array}}{(A \wedge B) \vee \Delta^T}\ \wedge I \qquad \frac{\vdots}{\dfrac{(A_1 \wedge A_2) \vee \Delta^T}{A_i \vee \Delta^T}\ \wedge_i E\ \ (i = 1, 2)}$$

$$\frac{\vdots}{\dfrac{A_i \vee \Delta^T}{(A_1 \vee A_2) \vee \Delta^T}\ \vee_i I\ \ (i = 1, 2)}$$

$$\frac{\begin{array}{ccc}\vdots & [x:A] & [y:B]\\ & \vdots & \vdots\\ (A \vee B) \vee \Delta^T & C \vee \Delta^T & C \vee \Delta^T\end{array}}{C \vee \Delta^T}\ (x, y) - \vee E$$

$$\frac{\begin{array}{c}[x:A]\\ \vdots\\ B\end{array}}{(A \to B) \vee \Delta^T}\ (x)- \to I \qquad \frac{\begin{array}{cc}\vdots & \vdots\\ (A \to B) \vee \Delta^T & A \vee \Delta^T\end{array}}{B \vee \Delta^T} \to E$$

$$\frac{\begin{array}{c}\vdots\\ A \vee \Delta^T(E^u)\end{array}}{(A \lhd E) \vee \Delta^T(\widehat{E^u})} \lhd I \qquad \frac{\begin{array}{c}\vdots\\ (A \lhd E) \vee \Delta^T(\widehat{E^u})\end{array}}{A \vee \Delta^T(E^u)} \lhd E$$

$$\frac{A \vee \Delta^T \vee \Delta^T}{A \vee \Delta^T}\ Contraction.$$

The A in the conclusion of *throw* is an arbitrary formula; the Δ^T in the conclusion of the $\to I$ is arbitrary, while the exceptional part in the premise of $\to I$ is empty.

Examples If $\Gamma \equiv (A \wedge B) \vee (C \vee D)$, disjunctive components of Γ are the following.

$$\Gamma, A \wedge B, C \vee D, C, D$$

$\Gamma(\widehat{C}) \equiv (A \wedge B) \vee D$.

2. If $\Gamma \equiv (A \wedge B) \vee ((A \vee C) \vee A)$, then disjuntive components of Γ are the following.

$$\Gamma, A \wedge B, (A \vee C) \vee A, A \vee C, A, C, A$$

Γ has two occurences of A as disjunctive formulas. The left most subformula A in Γ is not a disjunctive component of Γ. Let $\Gamma(A)$ indicate the center occurence of the subformula A in Γ, which is a disjunctive component. Then $\Gamma(\widehat{A}) \equiv (A \wedge B) \vee (C \vee A)$.

Definition 2.2. (Reduction)

We will define reductions of proofs of $\mathrm{NDJ}_{c/t}$, taking into account of the program execution. The reductions we employ are so-called the *inner most reductions*.

In the following, V, C, M will denote parts of proofs to which reductions no longer apply. M' will denote a simulation of M, details of which is

obvious from the expressions below.

$$\dfrac{\dfrac{\dfrac{\vdots\, V}{A \vee \Delta^T(\widehat{A^u})}\, throw}{A \vee \Delta^T(A^u)}\, catch}{A \vee \Delta^T(\widehat{A^u})} \quad \Longrightarrow \quad \dfrac{\vdots\, V}{A \vee \Delta^T}$$

$$\dfrac{\dfrac{\dfrac{\vdots\, V}{E \vee \Delta^T(\widehat{E^u})}\, throw}{A \vee \Delta^T(E^u)}}{\dfrac{\vdots\, C}{E \vee \Delta^T(E^u)}\, catch}{E \vee \Delta^T(\widehat{E^u})} \quad \Longrightarrow \quad \dfrac{\dfrac{\dfrac{\vdots\, V}{E \vee \Delta^T(\widehat{E^u})}}{E \vee \Delta^T(E^u)}\, throw}{E \vee \Delta^T(\widehat{E^u})}\, catch$$

$$\dfrac{\dfrac{\vdots\, V}{A \vee \Delta^T} \quad \dfrac{[x:A]}{\vdots\, M}}{C \vee \Delta^T}\, let \quad \Longrightarrow \quad \dfrac{\dfrac{\vdots\, V}{A \vee \Delta^T}}{\dfrac{\vdots\, M'}{C \vee \Delta^T \vee \Delta^T}}\, Contraction$$

$$\dfrac{\dfrac{\dfrac{[x:A]}{\vdots\, M}}{\dfrac{B}{(A \to B) \vee \Delta^T}}\, \to I \quad \dfrac{\vdots\, V}{A \vee \Delta^T}}{B \vee \Delta^T}\, \to E \quad \Longrightarrow \quad \dfrac{\dfrac{\vdots\, V}{A \vee \Delta^T}}{\vdots\, M'}{B \vee \Delta^T}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots\, V}{E \vee \Delta^T(\widehat{E^u})}\, throw}{A \vee \Delta^T(E^u)}}{\dfrac{\vdots\, C}{B \vee \Delta^T(E^u)}}{B \triangleleft E \vee \Delta^T(\widehat{E^u})}\, \triangleleft I \quad \Longrightarrow \quad \dfrac{\dfrac{\dfrac{\vdots\, V}{E \vee \Delta^T(\widehat{E^u})}}{B \vee \Delta^T(E^u)}\, throw}{B \triangleleft E \vee \Delta^T(\widehat{E^u})}\, \triangleleft I$$

$$\dfrac{\dfrac{\dfrac{\vdots\, V}{A \vee \Delta^T(E^u)}\, \triangleleft I}{A \triangleleft E \vee \Delta^T(\widehat{E^u})}\, \triangleleft E}{A \vee \Delta^T(E^u)} \quad \Longrightarrow \quad \dfrac{\vdots\, V}{A \vee \Delta^T(E^u)}$$

Notice that a common tag variable u occurs in the premise of $(\triangleleft I)$ and in the conclusion of $(\triangleleft E)$.

$$\dfrac{\dfrac{\dfrac{\vdots\ V}{E \vee \Delta^T(\widehat{E^u})}\ throw}{A \vee \Delta^T(E^u)}\ \triangleleft I}{A \triangleleft E \vee \Delta^T(\widehat{E^u})}\ \triangleleft E \qquad \Longrightarrow \qquad \dfrac{\vdots\ V}{E \vee \Delta^T(\widehat{E^v})}\ throw$$

Notice the change of tag variables. Since $\Delta^T(\widehat{E^u})$ is simply Δ^T, the change of tag variables from u to v causes no problem.

$$\dfrac{\dfrac{\vdots\ V_1 \quad \vdots\ V_2}{A_1 \vee \Delta^T \quad A_2 \vee \Delta^T}\ \wedge I}{(A_1 \wedge A_2) \vee \Delta^T}\ \wedge_i E \qquad \Longrightarrow \qquad \dfrac{\vdots\ V_i}{A_i \vee \Delta^T} \qquad (i = 1, 2)$$

$$\dfrac{\dfrac{\vdots\ V}{A_i \vee \Delta^T}\ \vee_i I}{(A_1 \vee A_2) \vee \Delta^T} \quad \dfrac{[x : A_1] \quad [y : A_2]}{\vdots\ M_1 \quad \vdots\ M_2} \quad C \vee \Delta^T \quad C \vee \Delta^T}{C \vee \Delta^T}\ \vee E \qquad \Longrightarrow$$

$$\dfrac{\dfrac{\vdots\ V}{A_i \vee \Delta^T}}{\dfrac{\vdots\ M_i'}{C \vee \Delta^T \vee \Delta^T}}\ Contraction}{C \vee \Delta^T} \qquad (i = 1, 2)$$

$$\dfrac{\dfrac{\dfrac{\vdots\ V}{A \vee E^u}\ weak}{A \vee E^u \vee E^u}\ Contraction}{A \vee E^u} \qquad \Longrightarrow \dfrac{\vdots\ V}{A \vee E^u}$$

Let Π denote a proof of $\mathrm{NDJ}_{c/t}$, and let $\Pi[\Pi']$ denote the fact that Π' is a subproof of Π. $\Pi[\Pi_1]$ is said to be reduced to $\Pi[\Pi_2]$ (written as $\Pi[\Pi_1] \Longrightarrow \Pi[\Pi_2]$) if $\Pi_1 \Longrightarrow \Pi_2$. In general, for proofs Π and Π^*, we write $\Pi \Longrightarrow \Pi^*$ when Π is reduced to Π^* in the sense above.

3. $\mathrm{NDJ}_{c/t}$ and term system

We will first introduce Nakano's terms and their reductions (cf. [4, 5, 6, 7]), and then give a term-representation of $\mathrm{NDJ}_{c/t}$ by these terms. A converse map will later be defined.

3.1. *Review: Nakano's terms and reductions*

Definition 3.1. Three mutually exclusive sets of symbols are assumed.

$Const$: a set of individual objects $c, d, \cdots$

Var : a countable set of individual variables $x, y, z, \cdots$

$Tvar$: a countable set of tag variables $u, v, w, \cdots$

Definition 3.2. Terms are defined as follows.

$$\begin{aligned} Term ::= {} & Const \mid Var \\ & \mid let\ Var\ =\ Term\ .\ Term \\ & \mid throw\ Tvar\ Term \mid catch\ Tvar\ Term \\ & \mid \lambda\ Var\ .\ Term^{*} \mid Term\ Term \\ & \mid \langle\ Term\ ,\ Term\ \rangle \mid proj_1\ Term \mid proj_2\ Term \\ & \mid inj_1\ Term \mid inj_2\ Term \mid case\ Term\ Var\ .\ Term\ Var\ .\ Term \\ & \mid \kappa\ Tvar\ .\ Term \mid Term\ Tvar \end{aligned}$$

Here $Term^{*}$ denotes a $Term$ which has no free occurrence of tag variables. $M, N, K, L, \cdots$ will be used to denote terms. A $Term$ which has no free occurrence of variables will be called a *closed term*.

Definition 3.3. In the following Val, the set of values, will be defined.

$$\begin{aligned} Val ::= {} & Const \mid Var \\ & \mid \lambda\ Var\ .\ Term \mid \langle\ Val\ ,\ Val\ \rangle \\ & \mid inj_1\ Val \mid inj_2\ Val \\ & \mid \kappa\ Tvar\ .\ Val \mid \kappa\ Tvar\ .\ throw\ Tvar\ Val \end{aligned}$$

$V, V', W, W', \cdots$ will be used for elements of Val.

Definition 3.4. The set of term-like expressions with an atom $*$, $Cont$, will be defined. A member of $Cont$ is called an *evaluation context*, and $*$ is called a *hole*.

$$\begin{aligned} Cont ::= {} & * \\ & \mid let\ Var\ =\ Cont\ .\ Term \\ & \mid throw\ Tvar\ Cont \mid catch\ Tvar\ Cont \\ & \mid Cont\ Term \mid Val\ Cont \\ & \mid \langle\ Cont\ ,\ Term\ \rangle \mid \langle\ Val\ ,\ Cont\ \rangle \mid proj_1\ Cont \mid proj_2\ Cont \\ & \mid inj_1\ Cont \mid inj_2\ Cont \mid case\ Cont\ Var\ .\ Term\ Var\ .\ Term \end{aligned}$$

$$\mid \kappa\ Tvar\ .\ Cont \mid Cont\ Tvar$$

The result of filling a hole $*$ with a term M in a context C will be denoted by $C[M]$.

Definition 3.5. *Call-by value rewriting* $\longmapsto$ is defined by the subsequent rules. $\mathcal{C}$ denotes any evaluation context which is not $*$ itself. V will denote a value . $M[V/x]$ denotes the result of substitution of V for x in M.

$$\begin{aligned}
catch\ u\ V &\longmapsto V \\
catch\ u\ \mathcal{C}[throw\ u\ V] &\longmapsto catch\ u\ \ (throw\ u\ V) \\
catch\ u\ \ [throw\ u\ V] &\longmapsto V \\
let\ x\ =\ V\ .\ M &\longmapsto M[V/x] \\
(\lambda\ x\ .\ M)\ V &\longmapsto M[V/x] \\
\kappa\ u\ .\ \mathcal{C}[throw\ u\ V] &\longmapsto \kappa\ u\ .\ throw\ u\ V \\
(\kappa\ u\ .\ V)\ v &\longmapsto V \\
(\kappa\ u\ .\ throw\ u\ V)\ v &\longmapsto throw\ v\ V \\
proj_1\ \langle V\ ,\ W\rangle &\longmapsto V \\
proj_2\ \langle V\ ,\ W\rangle &\longmapsto W \\
case\ (inj_1\ V)\ x\ .\ M\ y\ .\ N &\longmapsto M[V/x] \\
case\ (inj_2\ V)\ x\ .\ M\ y\ .\ N &\longmapsto N[V/y]
\end{aligned}$$

$\mathcal{C}[M]$ is said to be reduced to $\mathcal{C}[N]$ if $M \longmapsto N$, and this fact is written as $\mathcal{C}[M] \longmapsto \mathcal{C}[N]$.

A call-by value rewriting expresses the innermost computation.

3.2. *Assignment of terms to* $\mathbf{NDJ_{c/t}}$*-proofs*

We will express the proofs of $\mathrm{NDJ_{c/t}}$ in terms of the terms of $\mathcal{T}$.

Subsequently,

$$\begin{array}{c} \vdots\ \Pi \\ (t : C) \vee \Delta^T \end{array}$$

will represent that the term t is the computation process of deducing the main part C in the conclusion of the proof Π, that is, $C \vee \Delta^T$.

Proposition 3.1. *Let* Π *be an* $NDJ_{c/t}$*-proof (Definition 2.1), and let* $\tau(\Pi)$ *denote the assignment of a term from* $\mathcal{T}$ *to the main part of the conclusion of* Π *as defined below. Then* τ *preserves the reduction. Namely, suppose* $\Pi \Longrightarrow \Pi'$. $\tau(\Pi) \longmapsto \tau(\Pi')$ *if* $\tau(\Pi)$ *is not a value;* $\tau(\Pi) = \tau(\Pi')$ *if* $\tau(\Pi)$ *is a value;* $\tau(\Pi)$ *is a value if no reduction applies to* Π.

Let Π *be an* $NDJ_{c/t}$*-proof, and let* A *be the main part of the conclusion of* Π*. An assignment of a term* t *to* A *will be written in the form* $(t : A)$*, and with this* t*,* $\tau(\Pi) = t$*.* $(t : A)$ *will be defined subsequently.*

x *is assigned to*

$$[x : A] \ \textit{assumption},$$

that is, $(x : A)$*, or* $\tau([x : A]) = x$*. The label* x *is regarded as a term (variable) of deducing the assumption* A*.*

For the rules of inference, τ *is defined inductively.*

$$\frac{(u : A) \vee \Delta(\widehat{E^u})}{(u : A) \vee \Delta^T(E^u)}\ weak \qquad \frac{\begin{array}{cc}[x : A] & \\ \vdots & \vdots \\ (N : A) \vee \Delta^T & (M : C) \vee \Delta^T\end{array}}{(let\ x = N.M : C) \vee \Delta^T}\ let$$

$$\frac{\begin{array}{c}\vdots \\ (M : E) \vee \Delta^T(\widehat{E^u})\end{array}}{(throw\ u\ M : A) \vee \Delta^T(E^u)}\ throw \qquad \frac{\begin{array}{c}\vdots \\ (M : A) \vee \Delta^T(A^u)\end{array}}{(catch\ u\ M : A) \vee \Delta^T(\widehat{A^u})}\ catch$$

$$\frac{\begin{array}{c}[x : A] \\ \vdots \\ (M : B)\end{array}}{(\lambda\ x.M : A \to B) \vee \Delta^T}\ \to I \qquad \frac{\begin{array}{cc}\vdots & \vdots \\ (M : A \to B) \vee \Delta^T & (N : A) \vee \Delta^T\end{array}}{(MN : B) \vee \Delta^T}\ \to E$$

$$\frac{\begin{array}{cc}\vdots & \vdots \\ (M : A) \vee \Delta^T & (N : B) \vee \Delta^T\end{array}}{(\langle M, N\rangle : A \wedge B) \vee \Delta^T}\ \wedge I$$

$$\frac{\begin{array}{c}\vdots \\ (M : A \wedge B) \vee \Delta^T\end{array}}{(proj_1 M : A) \vee \Delta^T}\ \wedge_1 E \qquad \frac{\begin{array}{c}\vdots \\ (M : A \wedge B) \vee \Delta^T\end{array}}{(proj_2 M : B) \vee \Delta^T}\ \wedge_2 E$$

$$\frac{\begin{array}{c}\vdots \\ (M : A) \vee \Delta^T\end{array}}{(inj_1 M : A \vee B) \vee \Delta^T}\ \vee_1 I \qquad \frac{\begin{array}{c}\vdots \\ (M : B) \vee \Delta^T\end{array}}{(inj_2 M : A \vee B) \vee \Delta^T}\ \vee_2 I$$

$$\frac{\begin{array}{ccc}\vdots & [x : A] & [y : B] \\ & \vdots & \vdots \\ (L : A \vee B) \vee \Delta^T & (M : C) \vee \Delta^T & (N : C) \vee \Delta^T\end{array}}{(case\ L\ x.M\ y.N : C) \vee \Delta^T}\ \vee E$$

$$\frac{\begin{array}{c}\vdots\\(M:A)\vee\Delta^T(E^u)\end{array}}{(\kappa u.M:A\triangleleft E)\vee\Delta^T(\widehat{E^u})}\ \triangleleft I \qquad \frac{\begin{array}{c}\vdots\\(M:A\triangleleft E)\vee\Delta^T(\widehat{E^u})\end{array}}{(Mu:A)\vee\Delta^T(E^u)}\ \triangleleft E$$

$$\frac{\begin{array}{c}\vdots\\(M:C)\vee\Delta^T\vee\Delta^T\end{array}}{(M:C)\vee\Delta^T}\ Contraction$$

It is a routine work to prove the proposition.

3.3. *Assignment of proofs to terms*

Conversely, one can assign a proof to a term, which is to be shown subsequently.

Proposition 3.2. *Suppose first, a map α from variables to formulas is arbitrarily chosen: $\alpha(x)=A$.*

Then there is an assignment d of an $NDJ_{c/t}$-proof to a term relative to α such that d preserves reductions, that is, if $t\longmapsto s$, then $d(t)\Longrightarrow d(s)$.

d will be defined subsequently.

$$d(x)\equiv[x:A]\ \textit{if}\ \alpha(x)=A.$$

$$d(let\ x=N.M)\equiv\frac{\begin{array}{ccc}&&[x:A]\\d(N)&&d(M)\\A\vee\Delta^T&&C\vee\Delta^T\end{array}}{C\vee\Delta^T}\ (x)$$

$$d(throw\ u\ M)\equiv\frac{\begin{array}{c}d(M)\\E\vee\Delta^T(\widehat{E^u})\end{array}}{A\vee\Delta^T(E^u)} \qquad d(catch\ u\ M)\equiv\frac{\begin{array}{c}d(M)\\A\vee\Delta^T(A^u)\end{array}}{A\vee\Delta^T(\widehat{A^u})}$$

$$d(\lambda x.M)\equiv\frac{\begin{array}{c}[x:A]\\d(M)\\B\vee\Delta^T\end{array}}{(A\to B)\vee\Delta^T}\ (x) \qquad d(MN)\equiv\frac{\begin{array}{ccc}d(M)&&d(N)\\(A\to B)\vee\Delta^T&&A\vee\Delta^T\end{array}}{B\vee\Delta^T}$$

$$d(\langle M,N\rangle)\equiv\frac{\begin{array}{ccc}d(M)&&d(N)\\A\vee\Delta^T&&B\vee\Delta^T\end{array}}{(A\wedge B)\vee\Delta^T} \qquad d(proj_iM)\equiv\frac{\begin{array}{c}d(M)\\(A_1\wedge A_2)\vee\Delta^T\end{array}}{A_i\vee\Delta^T}$$

$$d(inj_i M) \equiv \dfrac{\begin{array}{c} d(M) \\ A_i \vee \Delta^T \end{array}}{(A_1 \vee A_2) \vee \Delta^T}$$

$$d(case\ L\ x.M\ y.N) \equiv \dfrac{\begin{array}{ccc} d(L) & [x:A] & [y:B] \\ (A \vee B) \vee \Delta^T & C \vee \Delta^T & C \vee \Delta^T \end{array}}{C \vee \Delta^T}\ (x,y)$$

$$d(\kappa u.M) \equiv \dfrac{\begin{array}{c} d(M) \\ A \vee \Delta^T(E^u) \end{array}}{(A \triangleleft E) \vee \Delta^T(\widehat{E^u})} \qquad d(Mu) \equiv \dfrac{\begin{array}{c} d(M) \\ (A \triangleleft E) \vee \Delta^T(\widehat{E^u}) \end{array}}{A \vee \Delta^T(E^u)}$$

The proposition can be proved by examining the definitions.

We now sum up our results.

Theorem 3.1. *The $NDJ_{c/t}$-proofs and $\mathcal{T}$-temrs are related as follows. There is a map from $NDJ_{c/t}$-proofs to $\mathcal{T}$-terms, $\tau(\Pi)$, in a manner that the computation of $\tau(\Pi)$ is reflected in the reduction of Π, and, if $\tau(\Pi)$ is a value, then the reduction of Π preserves the same value. Conversely, there is a map from $\mathcal{T}$ to $NDJ_{c/t}$, $d(t)$, in a manner that d preserves reductions.*

4. Realization of catch and throw: an example

Quoting an example from 1.2 of [7], we will explain how to realize the "catch and throw" mechanism in terms of the term assignments to $\mathrm{NDJ_{c/t}}$-proofs.

Let P be a program which, for a sequence of input data satisfying specifications $A_1, \cdots, A_n$, either outputs data satisfying a specification C or returns an error message E. Let us write this condition of P by:

$$P : A_1, \cdots, A_n \mapsto C; E$$

Let P^* denote a program which does the following job.
Given programs Q, R, S such that

$$Q : A_1, \cdots, A_n \mapsto D; E$$

$$R : A_1, \cdots, A_n, D \mapsto F;$$

$$S : A_1, \cdots, A_n, E \mapsto F;$$

P^* first runs Q. If the output satisfies D, then it proceeds to R, and if the output satisfies E, then it proceeds to S.

P^* can be represented in $\mathrm{NDJ}_{\mathrm{c/t}}$ as follows, where $[A]$ abbreviates $[A_1]\cdots[A_n]$. (We will omit the parentheses enclosing a term and the corresponding main part of a formula.)

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}[A]\\ \vdots\\ Q : D \vee E^u\end{matrix} \quad \dfrac{\begin{matrix}[A]\ [x : D]\\ \vdots\\ R : F\end{matrix}}{R : F \vee E^u}}{let\ x = Q.R : F \vee E^u}\ (x)}{throw\ v\ (let\ x = Q.R) : E \vee E^u \vee F^v}}{catch\ u\ (throw\ v\ (let\ x = Q.R)) : E \vee F^v} \quad \dfrac{\begin{matrix}[A]\ [y : E]\\ \vdots\\ S : F\end{matrix}}{throw\ v\ S : F \vee F^v}}{let\ y = (catch\ u\ (throw\ v\ (let\ x = Q.R))).(throw\ v\ S) : F \vee F^v}\ (y)}{catch\ v\ (let\ y = (catch\ u\ (throw\ v\ (let\ x = Q.R))).(throw\ v\ S)) : F}
$$

References

1. S. Hayashi, Surironrigaku(Mathematical Logic, in Japanes), Korona Pub.Co., 1989.
2. S.Hayashi, R.Sumitomo and K.Shii, *Towards animation of proofs-testing proofs by examples*, TCS 272, 177-195,2002.
3. Y.Kameyama and M.Sato, *Strong normalizability of the non-deterministic catch/throw calculi*, TCS 272, 223-245, 2002.
4. H.Nakano, A constructive formalization of the catch and throw mechanism, Proc. 7th LICS, pp.82-89, 1992.
5. H.Nakano, A constructive logic behind the catch and throw mechanism, APAL, 69, 269-301, 1994.
6. H.Nakano, The non-deterministic catch and throw mechanism and its subject reduction property, Logic, Language and Computation, LNCS 792, pp.61-72, Springer, 1994.
7. H. Nakano, Logical structures of the catch and throw mechanism, Ph.D Dissertation, University of Tokyo, 1995.
8. M.Nakata, Classical system NDK and its propositional subsystem $\mathrm{NDK}_{\mathrm{P}}^{\rightarrow}$, Master Thesis(in Japanese), Kyoto Sangyo University, 1998.
9. M.Nakata, N.Saneto and M.Yasugi, Classical and constructive propositional subsystems of NDK (in Japanese), The Bulletin of the ICS, Kyoto Sangyo University, vol.15, 37-81, 1999.
10. D. Prawitz, Natural Deduction, Almqvist and Wiksell, 1965.
11. N.Saneto, Constructive system NDJ_{Prop} and sstem $N_{c/t}$ which realizes catch and throw, Master Thesis(in Japanese), Kyoto Sangyo Univesity, 1998.
12. M.Yasugi and S.Hayashi, Interpretations of transfinite recursion and parametric abstraction in types, Words, Languages and Combinatorics II (1994),

452-464 (edited by M. Ito and H. Jurgensen), World Scientific Publ. Co., Singapore
13. M.Yasugi and M.Nakata, On natural proofs in NDK, Proceedings of RIMS(in Japanese), 976, 13-26, 1997.
14. M. Yasugi and M.Nakata, NDK and Natural Reasoning, Proceedings of the 6th ALC, 285-309, 1998
15. M. Yasugi and K. Ryu, NDK, A New Classical System, The Bull. of the Inst. of Computer Science of Kyoto Sangyo Univ., vol.11, 1-25, 1994.