Seminarium: logiki nieklasyczne, 28.11.2013r., Logika liniowa. Na podstawie artykułu Philipa Wadlera "Is there a use for linear logic?"(University of Glasgow, March 1991).

1 Motywacja dla wprowadzenia liniowego systemu typowania

Popatrzmy na reguły przypisywania termom typów w rachunku lambda w logice intuicjonistycznej.

$$\text{Id}\ \frac{}{x : U \vdash x : U} \qquad \text{Exchange}\ \frac{A, x : U, y : V, B \vdash w : W}{A, y : V, x : U, B \vdash w : W}$$

$$\text{Weakening}\ \frac{A \vdash v : V}{A,\ x : U \vdash v : V} \qquad \text{Contraction}\ \frac{A,\ x : U,\ y : U \vdash v : V}{A,\ z : U \vdash v^{z,z}_{x,y} : V}$$

$$\to\text{-I}\ \frac{A,\ x : U \vdash v : V}{A \vdash (\lambda x.\, v) : (U \to V)} \qquad \to\text{-E}\ \frac{A \vdash t : (U \to V) \qquad B \vdash u : U}{A,\ B \vdash (t\, u) : V}$$

$$\times\text{-I}\ \frac{A \vdash u : U \qquad B \vdash v : V}{A,\ B \vdash (u, v) : (U \times V)} \qquad \times\text{-E}\ \frac{A \vdash t : (U \times V) \qquad B,\ x : U,\ y : V \vdash w : W}{A, B \vdash (\text{case } t \text{ of } \{(x, y) \to w\}) : W}$$

$$+\text{-I}\ \frac{A \vdash u : U}{A \vdash (inl\ u) : (U + V)} \qquad \frac{A \vdash v : V}{A \vdash (inr\ v) : (U + V)}$$

$$+\text{-E}\ \frac{A \vdash t : (U + V) \qquad B,\ x : U \vdash u : W \qquad B,\ y : V \vdash v : W}{A, B \vdash (\text{case } t \text{ of } \{inl\, x \to u;\ inr\, y \to v\}) : W}$$

Pierwsze cztery reguły (Id, Exchange, Weakening, Contraction) to reguły strukturalne, odnoszące się do pojedynczych zmiennych. Reszta odnosi się do pozostałych termów. Jasne jest, że wymazując w powyższych regułach wszystkie zmienne, termy oraz traktując $\to$ jako implikację, $\times$ jako koniunkcję, a $+$ jak alternatywę, otrzymamy reguły logiki intuicjonistycznej. To przekształcenie jest znane jako izomorfizm Curry'ego-Howarda. Popatrzmy, jak zmienią się reguły strukturalne:

$$\text{Id}\ \frac{}{U \vdash U}, \quad \text{Exch}\ \frac{A,\ U,\ V,\ B \vdash W}{A,\ V,\ U,\ B \vdash W},$$

$$\text{Weak}\ \frac{A \vdash W}{A,\ U \vdash W}, \quad \text{Cont}\ \frac{A,\ U,\ U \vdash W}{A,\ U \vdash W}.$$

Reguła Weakening mówi, że jakaś wartość może zostać pominięta, natomiast Contraction, że można jej użyć dwukrotnie. Chcielibyśmy to ograniczyć.

2 Wprowadzenie spójnika modalnego "!"

Zastrzegamy, że reguły kontrakcji i osłabiania mogą być użyte tylko dla zmiennych typu $(!U)$. Te typy będziemy nazywać nieliniowymi, natomiast pozostałe liniowymi (liniowy system typowania obejmuje typy zarówno liniowe, jak i nieliniowe). Intuicja jest taka, że wartości typu nieliniowego mogą być dzielone, natomiast liniowego nie.

Ograniczamy się do spójników logiki liniowej: $\oplus, \otimes, !, \multimap$.
Gramatyka dla liniowych typów to:

$$T, U, C ::= X \mid (!U) \mid (U \multimap V) \mid (U \otimes V) \mid (U \oplus V).$$

Reguły liniowego systemu typowania są podobne do wcześniejszych, poza czterema odnoszącymi się do "!". Zobaczmy:

$$\text{Id}\ \frac{}{x : U \vdash x : U} \qquad \text{Exchange}\ \frac{A, x : U, y : V, B \vdash w : W}{A, y : V, x : U, B \vdash w : W}$$

$$\text{Promotion}\ \frac{(!A) \vdash v : V}{(!A) \vdash v : (!V)} \qquad \text{Dereliction}\ \frac{A,\ x : U \vdash v : V}{A,\ x : (!U) \vdash v : V}$$

$$\text{Weakening}\ \frac{A \vdash v : V}{A,\ x : (!U) \vdash v : V} \qquad \text{Contraction}\ \frac{A,\ x : (!U),\ y : (!U) \vdash v : V}{A,\ z : (!U) \vdash v^{z,z}_{x,y} : V}$$

$$\multimap\text{-I}\ \frac{A,\ x : U \vdash v : V}{A \vdash (\lambda x.\ v) : (U \multimap V)} \qquad \multimap\text{-E}\ \frac{A \vdash t : (U \multimap V) \qquad B \vdash u : U}{A,\ B \vdash (t\ u) : V}$$

$$\otimes\text{-I}\ \frac{A \vdash u : U \qquad B \vdash v : V}{A,\ B \vdash (u, v) : (U \otimes V)} \qquad \otimes\text{-E}\ \frac{A \vdash t : (U \otimes V) \qquad B,\ x : U,\ y : V \vdash w : W}{A, B \vdash (\text{case } t \text{ of } \{(x, y) \to w\}) : W}$$

$$\oplus\text{-I}\ \frac{A \vdash u : U}{A \vdash (inl\ u) : (U \oplus V)} \qquad \frac{A \vdash v : V}{A \vdash (inr\ v) : (U \oplus V)}$$

$$\oplus\text{-E}\ \frac{A \vdash t : (U \oplus V) \qquad B,\ x : U \vdash u : W \qquad B,\ y : V \vdash v : W}{A, B \vdash (\text{case } t \text{ of } \{inl\ x \to u;\ inr\ y \to v\}) : W}$$

Przez $!A$ rozumiemy listę założeń $x_1 : (!U_1), x_2 : (!U_2), \cdots, x_n : (!U_n)$
Reguła Promotion mówi, że wartość może być dzielona, jeśli tylko jej zmienne wolne moją być dzielone.
Trzy pozostałe reguły odnoszące się do "!"mówią o używaniu wartości typu $(!U)$: może być użyta dokładnie raz (Dereliction), wiele razy (Contraction), w ogóle nie być użyta (Weakening).

3 Wiele typów dla jednego termu

Okazuje się, że termowi może być przypisany więcej niż jeden typ. Chcielibyśmy, aby było tak. jak w sytermie intuicjonistycznym, że każdy term ma jeden najogólniejszy typ, z którego przez odpowiednie podstawienie można wyprowadzić pozostałe. Nie można tego zrobić zwyczajnie, bo jak popatrzymy na identyczność, to ma ona dwa typy: $\lambda x.x : X \multimap X$ oraz $\lambda x.x : !X \multimap X$. Widać, że żadnego z nich nie można uzystać z deugiego przez pewne podstawienie. Spróbujmy zmienić definicję najogólniejszego typu.

3.1 Podtypy

Definicja 3.1. U jest podtypem V ($U \leqslant V$), wtedy gdy term typu U będzie miał również typ V.

Uwaga: Ze względu na argument funkcji nie jest to monotoniczne, tzn. jeśli $U \leqslant U'$ oraz $V \leqslant V'$ to $U' \multimap V \leqslant U \multimap V'$.

Pytanie, które powinno się nam nasunąć, to czy powinniśmy przyjąć $U \leqslant (!U)$ czy $(!U) \leqslant U$? Jeśli popatrzymy na regułę Promotion, to jeśli mamy $v : V$ to również $v : !V$, co sugerowałoby przyjęcie $U \leqslant (!U)$.Natomiast po przeanalizowaniu reguły Dereliction i reguły $(\multimap I)$ to jeśli $\lambda x.v : U \multimap V$ to również ma typ $\lambda x.v : (!U) \multimap V$, co ze względu na powyższą uwagę sugeruje

wzięcie $(!U) \leqslant U$.

Rozważmy jednak następujący przykład: $\lambda f.\lambda x.\lambda y.f(x,y)$. Ten term może mieć, np. nastepujące typowania:

$$((X \otimes Y) \multimap Z) \multimap X \multimap Y \multimap Z$$
$$((X \otimes Y) \multimap Z) \multimap !X \multimap !Y \multimap Z$$
$$(!(X \otimes Y) \multimap Z) \multimap !X \multimap !Y \multimap Z$$
$$(!(!X \otimes !Y) \multimap Z) \multimap !X \multimap !Y \multimap Z$$

A nie może mieć np.:

$$(!(X \otimes Y) \multimap Z) \multimap X \multimap Y \multimap Z$$

Teraz niezależnie od tego, czy przyjmiemy $U \leqslant (!U)$ czy $(!U) \leqslant U$, nie znajdziemy typu, który jest 'większy' od wszystkich prawidłowych typowań, a nie jest większy od nieprawidłowego. Zatem definiowanie najogólniejszego typu dla termu poprzez postypy nie jest do końca dobrym pomysłem.

4 Typy użytkowe (Use types)

Uogólnimy pojęcie typu, by uwzględniało nie tylko zmienne typowe, ale także użytkowe, które będa wskazywać kiedy wartość zmiennej jest użyta w sposób liniowy, a kiedy w sposób nieliniowy. Dla zobrazowania popatrzmy na przykład. Typ funkcji identycznościowej zapiszemy jako $!^m X \multimap X$, gdzie m to zmienna użytkowa. W zależności od tego, czy przyjmiemy $m = 0$ czy $m = 1$, otrzymamy odpowiednio dwa przypadki typowań naszej funkcji: $X \multimap X$ oraz $!X \multimap X$.

4.1 Użytek, kontekst i lista użytków

Popatrzmy na to bardziej formalnie:

Definicja 4.1. Use (użytek) jest zmienną użytkową, 0 (oznaczając liniowość) lub 1 (oznaczając nieliniowość):

$$i, j, k ::= m \mid 0 \mid 1$$

Jak widać typ $(!U)$ uogólnia się do typu $(!^i U)$.

Definicja 4.2. Typ użytkowy (use type) ma jedną z form:

$$T, U, V ::= X \mid (!^i U) \mid (U \multimap V) \mid (U \otimes V) \mid (U \oplus V).$$

Definicja 4.3. Kontekst to zbiór nierówności między użytkami:

$$C, D, E ::= i_1 \geqslant j_1, i_2 \geqslant j_2, \cdots, i_n \geqslant j_n.$$

Podstawienie za każdą zmienną użytkową 1 lub 0 będzie albo spełniać kontekst albo nie. Nierówności postaci $i \geqslant 0$ lub $j \leqslant 1$ są spełnione zawsze, więc mogą zostać pominięte w zbiorze kontekstów. Oczywiście nierówność $i \geqslant 1$ może być spełniona tylko wtedy, gdzy $i = 1$, natomiast nierówność $j \leqslant 0$ tylko gdy $j = 0$. Mówimy, że kontekst jest niespełnialny, jeśli jego przechodnie domknięcie zawiera nierówność $0 \leqslant 1$.

Definicja 4.4. Use list (lista użytków) to zbiór takich par:

$$I, J ::= x_1 : i_1, \cdots, x_m : i_m$$

(x_i parami różne).

Jeśli A jest zbiorem założeń, a I listą użytków to zapis $!^I A$ oznacza zbiór założeń: $x_1 : !^{i_1} U_1, \cdots, x_n :!^{i_n} U_n$, natomiast zapis $I \geqslant j$ oznacza kontekst $i_1 \geqslant j, \cdots, i_n \geqslant j$.

Osąd (type judgement) przyjmuje postać: $A \vdash t : T[C]$ i mówimy, że przy założeniach A, term t ma typ T w kontekście C.
Popatrzmy, jak zmienią się reguły typowania:

$$\text{Id}\ \frac{}{x : U \vdash x : U\ [\,]} \qquad \text{Exchange}\ \frac{A, x : U, y : V, B \vdash w : W\ [C]}{A, y : V, x : U, B \vdash w : W\ [C]}$$

$$\text{Promotion}\ \frac{(!^I A) \vdash v : V\ [C]}{(!^I A) \vdash v : (!^j V)\ [C,\ I \geq j]} \qquad \text{Dereliction}\ \frac{A,\ x : U \vdash v : V\ [C]}{A,\ x : (!^i U) \vdash v : V\ [C]}$$

$$\text{Weakening}\ \frac{A \vdash v : V\ [C]}{A,\ x : (!^I U) \vdash v : V\ [C]} \qquad \text{Contraction}\ \frac{A,\ x : (!^I U),\ y : (!^I U) \vdash v : V\ [C]}{A,\ z : (!^I U) \vdash v^{z,z}_{x,y} : V\ [C]}$$

$$\multimap\text{-I}\ \frac{A,\ x : U \vdash v : V\ [C]}{A \vdash (\lambda x.\ v) : (U \multimap V)\ [C]} \qquad \multimap\text{-E}\ \frac{A \vdash t : (U \multimap V)\ [C] \qquad B \vdash u : U\ [D]}{A,\ B \vdash (t\ u) : V\ [C,\ D]}$$

$$\otimes\text{-I}\ \frac{A \vdash u : U\ [C] \qquad B \vdash v : V\ [D]}{A,\ B \vdash (u, v) : (U \otimes V)\ [C,\ D]}$$

$$\otimes\text{-E}\ \frac{A \vdash t : (U \otimes V)\ [C] \qquad B,\ x : U,\ y : V \vdash w : W\ [D]}{A, B \vdash (\text{case}\ t\ \text{of}\ \{(x, y) \to w\}) : W\ [C,\ D]}$$

$$\oplus\text{-I}\ \frac{A \vdash u : U\ [C]}{A \vdash (inl\ u) : (U \oplus V)\ [C]} \qquad \frac{A \vdash v : V\ [C]}{A \vdash (inr\ v) : (U \oplus V)\ [C]}$$

$$\oplus\text{-E}\ \frac{A \vdash t : (U \oplus V)\ [C] \qquad B,\ x : U \vdash u : W\ [D] \qquad B,\ y : V \vdash v : W\ [E]}{A, B \vdash (\text{case}\ t\ \text{of}\ \{inl\ x \to u;\ inr\ y \to v\}) : W\ [C,\ D,\ E]}$$

Przede wszystkim wszędzie dodajemy konteksty. Ponadto w regule Promotion, pojawia się w zbiorze kontekstu $J \geqslant j$, co zapewnia nam, że mamy typ z ! tylko wtedy, gdy każdy typ

w założeniach odpowiada typowi nieliniowemu.

4.2 Najbardziej ogólny typ

Powiedzmy teraz, co dokładnie oznacza, że term ma najbardziej ogóny typ. Na początku zdefiniujmy podstawienie.

Definicja 4.5. Podstawieniem nazwiemy funkcję, która przekształca typ zmiennej w typ użytkowy i zmienne użytkowe w użytki.

Podstawienie działa na założeniach, typach i kontekstach w zwykły sposób: Mając term t i dwa osądy dla niego:

$$A \vdash t : T[C], \quad A' \vdash t : T'[C']$$

mówimy, że drugi jest instancją pierwszego, jeśli istnieje podstawienie S, takie że: $SA = A'$, $ST = T'$ oraz SC jest podzbiorem przechodniego domknięcia zbioru C'.

Definicja 4.6. Typowanie jest główne dla termu t, jeśli każde inne typowanie, jest jego instancją.

Teoria zupełności mówi, że każdy typowalny term ma główne typowanie. Jet to zazwyczaj dowodzone przez podanie algorytmu rekonstrukcji typu, który zawodzi, jeśli term nie jest typowalny albo zwraca ten typ główny. Podamy taki algorytm, jednak nie dla użytkowego systemu typowania, a systemu standardowego.

5 Wprowadzenie standardowego systemu typowania

5.1 Translacja Girarda

Liniowy system precyzuje pojęcie typu, jednak nie typowalności. Wiemy, że każdy osąd w systemie intuicjonistycznym może być przekształcony na odpowiadający mu osąd w systemie liniowym i na odwrót. W jedną stronę jest to proste. W systemie liniowym wystarczy opuścić wszystkie wystepowania spójnika modalnego, zamienić $\multimap$ na implikację, $\oplus$ na $+$, a $\otimes$ na $\times$. W drugą stronę możemy użyć Translacji Girarda:

$$X^o = X$$

$$(U \to V)^o = (!U^o) \multimap V^o$$

$$(U \times V)^o = (!U^o) \otimes (!V^o)$$

(W standardowej translacji powinno być zamiast $\otimes$ spójnik &, ale my go nie używamy)

$$(U + V)^o = (!U^o) \oplus (!V^o)$$

Translacja działa na osądzie następująco:

$$(x_1 : U_1, \cdots, x_n : U_n \vdash v : V)^o = x_1 :!U_1^o, \cdots, x_n :!U_n^o \vdash v : V^o$$

W skrócie: $(A \vdash v : V)^o = (!A^o \vdash v : V^o)$. Każda zmienna ma typ nieliniowy, ale term sam w sobie nie ma. Jeśli osąd $A \vdash v : V$ wywodzi się z systemu intuicjonistycznego, to odpowiadający mu osąd $!A^o \vdash v : V^o$ można wyprowadzić z systemu liniowego, wystarczy tylko

zastosować translację do każdej reguły wyprowadzenia.

Ciekawa rzecz dzieję się w regule Id i eliminacji implikacji. Popatrzmy:

$$(x : U \vdash x : U)^o = x :!U^o \vdash x : U^o$$

To jest połączenie reguły Id i Dereliction.

$$\text{Der}\ \frac{\text{Id}\ \dfrac{}{x : U^\circ \vdash x : U^\circ}}{x :!U^\circ \vdash x : U^\circ.}$$

Natomiast reguła eliminacj implikacji przekształca się w kombnację reguł: Promotion i wprowadzenia liniowego wnioskowania:

$$\multimap\text{-E}\ \frac{!A^\circ \vdash t : (!U^\circ \multimap V^\circ) \qquad \text{Pro}\ \dfrac{!B^\circ \vdash u : U^\circ}{!B^\circ \vdash u :!U^\circ}}{!A^\circ, !B^\circ \vdash (t\ u) : V^\circ.}$$

To kodowanie nie minimalizuje liczby modalnych spójników, np. funkcja identycznościowa $\lambda x.x$ ma typ intuicjonistyczny $X \to X$ i to koduje się na $!X \multimap X$, a wiemy, że term ten ma również typ $X \multimap X$, który nie zawiera żadnego spójnika modalnego. Mimo, że to kodowanie nie minimalizuje nam liczby spójników modalnych, to gwarantuje, że typowalny term posiada typowanie, które nie zawiera "!"w pewnych miejscach. (Głównie popatrzmy na funkcje: nie może mieć ona wartości typu nieliniowego). To motywuje do zdfiniowania standardowych liniowych typów.

5.2 Typy standardowe (standard types)

Definicja 5.1. Standard types

$$T, U, V ::= X \mid (!^i U \multimap V) \mid (!^i U \otimes !^j V) \mid (!^i U \oplus !^j V)$$

Osąd ma formę $!^I A \vdash t : T[C]$, gdzie T i wszystkie typy w A są standardowe.

Wyprowadzenie standardowego typowania może 'po drodze' zawierać jakieś typowania, które nie są stadardowe. W szczególności użycie Dereliction Rule lub Promotion Rule będzie dawało hipotezę lub konkluzję nie w standardowej formie. Dlatego te reguły zostały pominięte dla tego systemu typowania i w pewnym sensie wbudowane w reguły Id i eliminacji wnioskowania.

$$\text{Id}\ \frac{}{x :!^i U \vdash x : U\ []} \qquad \text{Exchange}\ \frac{!^I A,\ x :!^i U,\ y :!^j V,\ !^J B \vdash w : W\ [C]}{!^I A,\ y :!^j V,\ x :!^i U,\ !^J B \vdash w : W\ [C]}$$

$$\text{Weakening}\ \frac{!^I A \vdash v : V\ [C]}{!^I A,\ x :!^I U \vdash v : V\ [C]} \qquad \text{Contraction}\ \frac{!^I A,\ x :!^I U,\ y :!^I U \vdash v : V\ [C]}{!^I A,\ z :!^I U \vdash v^{z,z}_{x,y} : V\ [C]}$$

$$\multimap\text{-I}\ \frac{!^I A,\ x : !^i U \vdash v : V\ [C]}{!^I A \vdash (\lambda x.\ v) : (!^i U \multimap V)\ [C]}$$

$$\multimap\text{-E}\ \frac{!^I A \vdash t : (!^i U \multimap V)\ [C] \qquad !^J B \vdash u : U\ [D]}{!^I A,\ !^J B \vdash (t\ u) : V\ [C,\ D,\ J \geq i]}$$

$$\otimes\text{-I}\ \frac{!^I A \vdash u : U\ [C] \qquad !^J B \vdash v : V\ [D]}{!^I A,\ !^J B \vdash (u, v) : (!^i U \otimes !^j V)\ [C,\ D,\ I \geq i,\ J \geq j]}$$

$$\otimes\text{-E}\ \frac{!^I A \vdash t : (!^i U \otimes !^j V)\ [C] \qquad !^J B,\ x : !^i U,\ y : !^j V \vdash w : W\ [D]}{!^I A,\ !^J B \vdash (\text{case } t \text{ of } \{(x, y) \to w\}) : W\ [C,\ D]}$$

$$\oplus\text{-I}\ \frac{!^I A \vdash u : U\ [C]}{!^I A \vdash (inl\ u) : (!^i U \oplus !^j V)\ [C,\ I \geq i]} \qquad \frac{!^J B \vdash v : V\ [D]}{!^J B \vdash (inr\ v) : (!^i U \oplus !^j V)\ [D,\ J \geq j]}$$

$$\oplus\text{-E}\ \frac{!^I A \vdash t : (!^i U \oplus !^j V)\ [C] \qquad !^J B,\ x : !^i U \vdash u : W\ [D] \qquad !^J B,\ y : !^j U \vdash v : W\ [E]}{!^I A,\ !^J B \vdash (\text{case } t \text{ of } \{inl\ x \to u;\ inr\ y \to v\}) : W\ [C,\ D,\ E]}$$

6 Rekonstrukcja typu

Dla danego termu t, jeśli istnieje typowanie dla t, to wywołanie L(t) zwróci trójkę $(!^I A, T, C)$, taką, że $!^I A \vdash t : T[C]$ jest głównym standardowym typowaniem dla t (tzn. jeden, który jest instancją wszystkich innych standardowych typowań dla t) i zawiedzie w przeciwnym przypadku. Algorytm rekonstrukcji dla typowania standardowego:

$$\begin{array}{lcl} L(x) & = & \text{let} \quad m, X \text{ be fresh} \\ & & \text{in} \quad (x : !^m X;\ X;\ ) \end{array}$$

$$\begin{array}{lcl} L(\lambda x.\ v) & = & \text{let} \quad (!^I A,\ x : !^i U;\ V;\ C) = L(v) \\ & & \text{in} \quad (!^I A;\ (!^i U \multimap V);\ C) \end{array}$$

$$
\begin{array}{lcll}
L(t\,u) & = & \text{let} & m, Y \text{ be fresh} \\
 & & & (!^I A;\ T;\ C) = L(t) \\
 & & & (!^J B;\ U;\ D) = L(u) \\
 & & & S = \text{unify}\,(T = (!^m U \multimap Y), !^I A = !^J B) \\
 & & \text{in} & (S!^I A,\ S!^J B;\ SY;\ SC,\ SD,\ SJ \geq Sm)
\end{array}
$$

$$
\begin{array}{lcll}
L((u, v)) & = & \text{let} & m, n \text{ be fresh} \\
 & & & (!^I A;\ U;\ C) = L(u) \\
 & & & (!^J B;\ V;\ D) = L(v) \\
 & & & S = \text{unify}\,(!^I A = !^J B) \\
 & & \text{in} & (S!^I A,\ S!^J B;\ (!^m SU \otimes !^n SV);\ SC,\ SD,\ SI \geq m,\ SJ \geq n)
\end{array}
$$

$$
\begin{array}{l}
L(\text{case } t \text{ of } \{(x, y) \to w\}) \\
\qquad \begin{array}{lcll}
 & = & \text{let} & (!^I A;\ T;\ C) = L(t) \\
 & & & (!^J B,\ x : !^i U,\ y : !^j V;\ W;\ D) = L(w) \\
 & & & S = \text{unify}\,(T = (!^i U \otimes !^j V),\ !^I A = !^J B) \\
 & & \text{in} & (S!^I A, S!^J B;\ SW;\ SC,\ SD)
\end{array}
\end{array}
$$

$$
\begin{array}{lcll}
L(inl\ u) & = & \text{let} & m, n, Y \text{ be fresh} \\
 & & & (!^I A;\ U;\ C) = L(u) \\
 & & \text{in} & (!^I A;\ !^m U \oplus !^b Y;\ C,\ I \geq m)
\end{array}
$$

$$
\begin{array}{lcll}
L(inr\ v) & = & \text{let} & m, n, X \text{ be fresh} \\
 & & & (!^J B;\ V;\ D) = L(v) \\
 & & \text{in} & (!^J B;\ !^m X \oplus !^n V;\ D,\ J \geq n)
\end{array}
$$

$$
\begin{array}{l}
L(\text{case } t \text{ of } \{inl\, x \to u;\ inr\, y \to v\}) \\
\qquad \begin{array}{lcll}
 & = & \text{let} & (!^I A;\ T;\ C) = L(t) \\
 & & & (!^J B,\ x : !^i U;\ W;\ D) = L(u) \\
 & & & (!^{J'} B',\ y : !^j V;\ W';\ D') = L(v) \\
 & & & S = \text{unify}(\ T = (!^i U \oplus !^j V),\ W = W', \\
 & & & \qquad\qquad !^I A = !^J B,\ !^I A = !^{J'} B',\ J \setminus J',\ J' \setminus J) \\
 & & \text{in} & (S!^I A, S!^J B, S!^{J'} B';\ SW;\ SC,\ SD,\ SD')
\end{array}
\end{array}
$$