Computation and Deduction

Lecture 16: The Curry-Howard Isomorphism
March 6, 1997

1. Natural Deduction
2. A Simply-Typed λ-Calculus
3. The Curry-Howard Isomorphism

Natural Deduction I

```
nd : o -> type.  % deductions
%name nd D E

andi    : nd A -> nd B -> nd (A and B).
andel   : nd (A and B) -> nd A.
ander   : nd (A and B) -> nd B.

impi    : (nd A -> nd B) -> nd (A imp B).
impe    : nd (A imp B) -> nd A -> nd B.

oril    : nd A -> nd (A or B).
orir    : nd B -> nd (A or B).
ore     : nd (A or B) -> (nd A -> nd C) -> (nd B -> nd C)
            -> nd C.

noti    : ({p:o} nd A -> nd p) -> nd (not A).
note    : nd (not A) -> {C:o} nd A -> nd C.
```

Natural Deduction II

```
truei    : nd (true).

falsee   : nd (false) -> nd C.

foralli : ({a:i} nd (A a)) -> nd (forall A).
foralle : nd (forall A) -> {T:i} nd (A T).

existsi : {T:i} nd (A T) -> nd (exists A).
existse : nd (exists A) -> ({a:i} nd (A a) -> nd C) -> nd C.
```

Simply-Typed λ-Calculus

```
tm : type.  %name tm M N

pair : tm -> tm -> tm.
fst  : tm -> tm.
snd  : tm -> tm.

lam  : o -> (tm -> tm) -> tm.
app  : tm -> tm -> tm.

inl  : o -> tm -> tm.
inr  : o -> tm -> tm.
case : tm -> (tm -> tm) -> (tm -> tm) -> tm.

mu   : o -> (o -> tm -> tm) -> tm.
mapp : o -> tm -> tm -> tm.
triv : tm.
abort : o -> tm -> tm.
```

Typing Rules I

```
# : tm -> o -> type. %name # P Q
%infix none 9 #

#pair : M # A
     -> N # B
     -> (pair M N) # (A and B).

#fst  : M # (A and B)
     -> fst M # A.

#snd  : M # (A and B)
     -> snd M # B.

#lam  : ({u:tm} u # A -> (M u) # B)
     -> (lam A M) # (A imp B).
```

Typing Rules II

```
#app  : M # (A imp B)
     -> N # A
     -> (app M N) # B.

#inl  : M # A
     -> (inl B M) # (A or B).

#inr  : N # B
     -> (inr A N) # (A or B).

#case : M # (A or B)
     -> ({u1:tm} u1 # A -> (N1 u1) # C)
     -> ({u2:tm} u2 # B -> (N2 u2) # C)
     -> (case M N1 N2) # C.
```

Typing Rules III

```
#mu    : ({p:o}{u:tm} u # A -> (M p u) # p)
     -> (mu A M) # (not A).

#mapp : M # (not A)
     -> N # A
     -> (mapp C M N) # C.

#triv : triv # true.

#abort : M # false
      -> (abort C M) # C.
```

Curry-Howard Isomorphism I

```
ch : nd A -> {M:tm} M # A -> type.

ch_andi : ch (andi D E) (pair M N) (#pair P Q)
           <- ch D M P
           <- ch E N Q.

ch_andel : ch (andel D) (fst M) (#fst P)
            <- ch D M P.

ch_ander : ch (ander D) (snd M) (#snd P)
            <- ch D M P.
```

Curry-Howard Isomorphism II

```
ch_impi  : ch (impi D) (lam A M) (#lam P)
            <- ({u : nd A} {x : tm} {u' : x # A}
                  ch u x u' -> ch (D u) (M x) (P x u')).

ch_impe  : ch (impe D E) (app M N) (#app P Q)
            <- ch D M P
            <- ch E N Q.

ch_oril   : ch (oril D) (inl B M) (#inl P)
             <- ch D M P.

ch_orir   : ch (orir E) (inr A N) (#inr Q)
             <- ch E N Q.
```

Curry-Howard Isomorphism III

```
ch_ore    : ch (ore D E1 E2) (case M N1 N2) (#case P Q1 Q2)
             <- ch D M P
             <- ({u1:nd A} {x1:tm} {u1' : x1 # A}
                   ch u1 x1 u1'
                   -> ch (E1 u1) (N1 x1) (Q1 x1 u1'))
             <- ({u2:nd B} {x2:tm} {u2' : x2 # B}
                   ch u2 x2 u2'
                   -> ch (E2 u2) (N2 x2) (Q2 x2 u2')).

ch_truei  : ch (truei) (triv) (#triv).

ch_falsee : ch (falsee D) (abort C M) (#abort P)
             <- ch D M P.
```