# Computation and Deduction

Lecture 20: Uniform Derivations
April 1, 1997

1. Uniform Derivations
2. Soundness of Uniform Derivations
3. Canonical Natural Deductions
4. Completeness of Uniform Derivations

# Uniform Derivations

```
solve  : o -> type.        % solve goal formulas
assume : o -> type.        % assume program formulas
>>     : o -> p -> type.   % immediate entailment
%infix none 8 >>
s_and  : solve (A1 and A2)
          <- solve A1
          <- solve A2.

s_imp  : solve (A2 imp A1)
          <- (assume A2 -> solve A1).

s_true : solve (true).

s_atom : solve (atom P)
          <- assume A
          <- A >> P.
```

# Immediate Entailment

```
s_atom : solve (atom P)
          <- assume A
          <- A >> P.

i_andl : A1 and A2 >> P
           <- A1 >> P.
i_andr : A1 and A2 >> P
           <- A2 >> P.

i_imp  : A2 imp A1 >> P
          <- A1 >> P
          <- solve A2.

i_atom : (atom P) >> P.
```

# Soundness of Uniform Derivations I

```
s_sound : solve A -> pf A -> type.
h_sound : assume A -> pf A -> type.
i_sound : A >> P -> (pf A -> pf (atom P)) -> type.

ss_and  : s_sound (s_and S2 S1) (andi D2 D1)
           <- s_sound S1 D1
           <- s_sound S2 D2.

ss_imp  : s_sound (s_imp S1) (impi D1)
           <- ({d:assume A} {u:pf A}
               h_sound d u -> s_sound (S1 d) (D1 u)).
```

# Soundness of Uniform Derivations II

```
ss_atom : s_sound (s_atom I2 H1) (D2 D1)
           <- h_sound H1 D1
           <- i_sound I2 D2.

is_andl : i_sound (i_andl I1)
                  ([u:pf (A1 and A2)] D1 (andel u))
           <- i_sound I1 D1.

is_imp  : i_sound (i_imp S2 I1)
                  ([u:pf (A2 imp A1)] D1 (impe u D2))
           <- i_sound I1 D1
           <- s_sound S2 D2.

is_atom : i_sound (i_atom) ([u:pf (atom P)] u).
```

# Canonical Natural Deductions I

```
can : {A:o} pf A -> type.  % Canonical deductions
atm : pf A -> type.        % Atomic deductions

can_andi : can (A and B) (andi D E)
            <- can A D
            <- can B E.

can_impi : can (A imp B) (impi D)
            <- ({u:pf A} atm u -> can B (D u)).

can_truei : can (true) (truei).
```

# Canonical Natural Deductions II

```
can_atm : can (atom P) D <- atm D.

% no atm_true
atm_andel : atm (andel D) <- atm D.
atm_ander : atm (ander D) <- atm D.

atm_impe : atm (impe D (E : pf B))
            <- atm D
            <- can B E.
```

# Completeness of Uniform Derivations

```
cmpcs : can A D -> solve A -> type.
cmpai : {D:pf A} atm D
          -> ({P:p} A >> P -> solve (atom P))
          -> type.

cmpcs_andi : cmpcs (can_andi CN2 CN1) (s_and S2 S1)
              <- cmpcs CN1 S1
              <- cmpcs CN2 S2.

cmpcs_impi : cmpcs (can_impi CN1) (s_imp S1)
              <- ({u:pf A2} {a:atm u} {d:assume A2}
                   cmpai u a ([P:p] [i:A2 >> P] s_atom i d)
                   -> cmpcs (CN1 u a) (S1 d)).
```