CIS 500
Software Foundations
Fall 2003

19 November (continued)
Meets and Joins
Adding Booleans

Suppose we want to add booleans and conditionals to the language we have been discussing.

For the **declarative** presentation of the system, we just add in the appropriate syntactic forms, evaluation rules, and typing rules.

\[
\Gamma \vdash \text{true} : \text{Bool} \quad \text{(T-TRUE)}
\]

\[
\Gamma \vdash \text{false} : \text{Bool} \quad \text{(T-FALSE)}
\]

\[
\Gamma \vdash t_1 : \text{Bool} \quad \Gamma \vdash t_2 : T \quad \Gamma \vdash t_3 : T
\]

\[
\frac{}{\Gamma \vdash \text{if } t_1 \text{ then } t_2 \text{ else } t_3 : T} \quad \text{(T-IF)}
\]
A Problem with Conditional Expressions

For the algorithmic presentation of the system, however, we encounter a little difficulty.

What is the minimal type of

\[
\text{if true then } \{x=\text{true}, y=\text{false}\} \text{ else } \{x=\text{true}, z=\text{true}\}
\]

?
The Algorithmic Conditional Rule

More generally, we can use subsumption to give an expression

\[
\text{if } t_1 \text{ then } t_2 \text{ else } t_3
\]

any type that is a possible type of both \( t_2 \) and \( t_3 \).

So the minimal type of the conditional is the least common supertype (or join) of the minimal type of \( t_2 \) and the minimal type of \( t_3 \).

\[
\Gamma \vdash t_1 : \text{Bool} \quad \Gamma \vdash t_2 : T_2 \quad \Gamma \vdash t_3 : T_3
\]

\[
\Gamma \vdash \text{if } t_1 \text{ then } t_2 \text{ else } t_3 : T_2 \lor T_3
\]

(T-IF)
The Algorithmic Conditional Rule

More generally, we can use subsumption to give an expression

\[ \text{if } t_1 \text{ then } t_2 \text{ else } t_3 \]

any type that is a possible type of both \( t_2 \) and \( t_3 \).

So the minimal type of the conditional is the least common supertype (or join) of the minimal type of \( t_2 \) and the minimal type of \( t_3 \).

\[
\frac{\Gamma \vdash t_1 : \text{Bool} \quad \Gamma \vdash t_2 : T_2 \quad \Gamma \vdash t_3 : T_3}{\Gamma \vdash \text{if } t_1 \text{ then } t_2 \text{ else } t_3 : T_2 \lor T_3} \quad \text{(T-IF)}
\]

Does such a type exist for every \( T_2 \) and \( T_3 \)??

CIS 500, 19 November (continued)
Existence of Joins

**Theorem:** For every pair of types $S$ and $T$, there is a type $J$ such that

1. $S \leq J$
2. $T \leq J$
3. If $K$ is a type such that $S \leq K$ and $T \leq K$, then $J \leq K$.

I.e., $J$ is the smallest type that is a supertype of both $S$ and $T$. 
Examples

What are the joins of the following pairs of types?

1. \{x:Bool,y:Bool\} and \{y:Bool,z:Bool\}?

2. \{x:Bool\} and \{y:Bool\}? 

3. \{x:{a:Bool,b:Bool}\} and \{x:{b:Bool,c:Bool}, y:Bool\}? 

4. {} and Bool? 

5. \{x:{}\} and \{x:Bool\}? 

6. Top→\{x:Bool\} and Top→\{y:Bool\}? 

7. \{x:Bool\}→Top and \{y:Bool\}→Top?
Meets

To calculate joins of arrow types, we also need to be able to calculate **meets** (greatest lower bounds)!

Unlike joins, meets do not necessarily exist.

E.g., `Bool → Bool` and `[]` have **no** common subtypes, so they certainly don’t have a greatest one!

However...
Existence of Meets

**Theorem:** For every pair of types $S$ and $T$, if there is any type $N$ such that $N <: S$ and $N <: T$, then there is a type $M$ such that

1. $M <: S$
2. $M <: T$
3. If $O$ is a type such that $O <: S$ and $O <: T$, then $O <: M$.

I.e., $M$ (when it exists) is the largest type that is a subtype of both $S$ and $T$.

**Jargon:** In the simply typed lambda calculus with subtyping, records, and booleans...

- The subtype relation has **joins**
- The subtype relation has **bounded** meets
Examples

What are the meets of the following pairs of types?

1. \{x: \text{Bool}, y: \text{Bool}\} \text{ and } \{y: \text{Bool}, z: \text{Bool}\}?

2. \{x: \text{Bool}\} \text{ and } \{y: \text{Bool}\}?

3. \{x: \{a: \text{Bool}, b: \text{Bool}\}\} \text{ and } \{x: \{b: \text{Bool}, c: \text{Bool}\}, y: \text{Bool}\}?

4. \{\}\text{ and } \text{Bool}?

5. \{x: \{\}\} \text{ and } \{x: \text{Bool}\}?

6. \text{Top} \rightarrow \{x: \text{Bool}\} \text{ and } \text{Top} \rightarrow \{y: \text{Bool}\}?

7. \{x: \text{Bool}\} \rightarrow \text{Top} \text{ and } \{y: \text{Bool}\} \rightarrow \text{Top}?
Calculating Joins

\[ S \lor T = \begin{cases} 
    \text{Bool} & \text{if } S = T = \text{Bool} \\
    M_1 \rightarrow J_2 & \text{if } S = S_1 \rightarrow S_2 \quad T = T_1 \rightarrow T_2 \\
    \{ j_1 : J_1 \left| 1 \leq j_1 \leq q \right. \} & \text{if } S = \{ k_j : S_j \left| 1 \leq j \leq m \right. \} \quad T = \{ l_i : T_i \left| 1 \leq i \leq n \right. \} \\
    \{ j_1 \left| 1 \leq j_1 \leq q \right. \} = \{ k_j \left| 1 \leq j \leq m \right. \} \cap \{ l_i \left| 1 \leq i \leq n \right. \} \\
    S_j \lor T_i = J_1 & \text{for each } j_1 = k_j = l_i \\
    \text{Top} & \text{otherwise}
\]
Calculating Meets

\[
S \land T = \begin{cases} 
S & \text{if } T = \text{Top} \\
T & \text{if } S = \text{Top} \\
\text{Bool} & \text{if } S = T = \text{Bool} \\
J_1 \rightarrow M_2 & \text{if } S = S_1 \rightarrow S_2 \quad T = T_1 \rightarrow T_2 \\
S_1 \lor T_1 = J_1 & S_2 \land T_2 = M_2 \\
\{m_l : M_l \mid l \in 1..q\} & \text{if } S = \{k_j : S_j \mid j \in 1..m\} \\
T = \{l_i : T_i \mid i \in 1..n\} \\
\{m_l \mid l \in 1..q\} = \{k_j \mid j \in 1..m\} \cup \{l_i \mid i \in 1..n\} \\
S_j \land T_i = M_l & \text{for each } m_l = k_j = l_i \\
M_l = S_j & \text{if } m_l = k_j \text{ occurs only in } S \\
M_l = T_i & \text{if } m_l = l_i \text{ occurs only in } T \\
\text{fail} & \text{otherwise}
\end{cases}
\]