# Solving Type Inference Problems

This document attempts to record more carefully the sequence of steps done "live" in Lecture 7b. Let's spell out a sequence of steps that will solve the set of type constraints generated from the example on slide 8 of that lecture.

**You are not required to follow this approach in your homework solutions!** It is probably easier to just "eyeball" a solution to smaller problems. This document is just for completeness.

Here are the constraints, numbered for easy reference:

(1$a$) $t_f = t_2$
(1$b$) $t_1 = t_7$
(2) $t_2 = t_x \to t_3$
(3$a$) $t_4 = \texttt{bool}$
(3$b$) $t_3 = t_5$
(3$c$) $t_3 = t_6$
(4) $t_4 = t_x$
(5) $t_5 = t_p$
(6) $t_5 = t_q$
(7$a$) $t_7 = \texttt{int}$
(7$b$) $t_8 = \texttt{int}$
(7$c$) $t_9 = \texttt{int}$
(8) $t_8 = \texttt{int}$
(9) $t_{10} = t_{11} \to t_9$
(10) $t_{10} = t_f$
(11) $t_{11} = t_r$

A *solution* to this constraint problem defines the type of each variable in terms of the *ground types* (i.e. variable-free types) `int`, `bool`, and $\to$. This solution can be viewed as a *substitution* from variables to types: if we apply the solution substitution, then all the constraints turn into vacuous equalities, like `bool = bool`.

To find such a solution, we use a *unification algorithm.* We process equations one at a time. Depending on the *shape* of the equation being processed, either this produces a *substitution* for some variable, which we apply to all other equations in the problem, or it produces new equations to add to the problem. In any case, we mark the equation as having been processed, so that we don't revisit it.

There are three shapes of equations to consider:

- var/nonvar: $v = t$ or $t = v$ for some $t$ that is not a variable, i.e. $t$ has the form `bool` or `int` or $d \to r$ (for some types $d, r$). In this case, we substitute $t$ for $v$. One subtlety: if $v$ appears in $t$, our constraint system is *circular*, and does not have a solution with finite types, so we terminate with an error.
- var/var: $v_1 = v_2$. In this case, we substitute one variable for the other (it doesn't matter which). If the vars are the same, there is nothing to do (and we can drop the equation altogether).

- nonvar/nonvar $t_1 = t_2$ where neither $t_1$ nor $t_2$ is a variable. In this case, if $t_1$ and $t_2$ are both `int` or `bool`, there is nothing to do (and we can drop the equation altogether). If $t_1 = d_1 \to r_1$ and $t_2 = d_2 \to r_2$, then we add the equations $d_1 = d_2$ and $r_1 = r_2$ to our problem. Finally, if $t_1$ and $t_2$ are *different* types, we terminate with an error: our constraint problem doesn't have a solution.

The order in which we process equations doesn't really matter. We mark processed equations with a ! sign. Note that we will continue to do substitutions in these marked equations as we proceed; this is crucial because, even though we never visit the marked equations again while solving, we will ultimately read off the solution from them

So here's how things might go. First, let's eliminate all var/nonvar equations. Initially, there are seven of them. Starting (somewhat at random) with $(3a)$, we substitute `bool` for $t_4$ throughout the problem (i.e. in (4)), obtaining this (slightly) simpler problem. .

$$\begin{array}{rl}
(1a) & t_f = t_2 \\
(1b) & t_1 = t_7 \\
(2) & t_2 = t_x \to t_3 \\
(!3a) & t_4 = \texttt{bool} \\
(3b) & t_3 = t_5 \\
(3c) & t_3 = t_6 \\
(4) & \texttt{bool} = t_x \\
(5) & t_5 = t_p \\
(6) & t_5 = t_q \\
(7a) & t_7 = \texttt{int} \\
(7b) & t_8 = \texttt{int} \\
(7c) & t_9 = \texttt{int} \\
(8) & t_8 = \texttt{int} \\
(9) & t_{10} = t_{11} \to t_9 \\
(10) & t_{10} = t_f \\
(11) & t_{11} = t_r
\end{array}$$

Repeating the process for $(7a), (7b), (7c)$ leads to:

| | |
|---|---|
| $(1a)$ | $t_f = t_2$ |
| $(1b)$ | $t_1 = \texttt{int}$ |
| $(2)$ | $t_2 = t_x \to t_3$ |
| $(!3a)$ | $t_4 = \texttt{bool}$ |
| $(3b)$ | $t_3 = t_5$ |
| $(3c)$ | $t_3 = t_6$ |
| $(4)$ | $\texttt{bool} = t_x$ |
| $(5)$ | $t_5 = t_p$ |
| $(6)$ | $t_5 = t_q$ |
| $(!7a)$ | $t_7 = \texttt{int}$ |
| $(!7b)$ | $t_8 = \texttt{int}$ |
| $(!7c)$ | $t_9 = \texttt{int}$ |
| $(8)$ | $\texttt{int} = \texttt{int}$ |
| $(9)$ | $t_{10} = t_{11} \to \texttt{int}$ |
| $(10)$ | $t_{10} = t_f$ |
| $(11)$ | $t_{11} = t_r$ |

Note that (8), which was originally one of the var/nonvar equations, has now become a nonvar/nonvar equation as a result of substitution. Since it has identical left and right-hand sides, processing it has no effect: in fact, we can drop it altogether, since it no longer contains useful information. On the other hand, we have gained two new var/no-var equations, namely $(1b)$ and $(4)$. Processing them leads to the following state (note that applying the substitution generated by $(1b)$ doesn't change anything, since $t_1$ doesn't appear anywhere else in the problem):

| | |
|---|---|
| $(1a)$ | $t_f = t_2$ |
| $(!1b)$ | $t_1 = \texttt{int}$ |
| $(2)$ | $t_2 = \texttt{bool} \to t_3$ |
| $(!3a)$ | $t_4 = \texttt{bool}$ |
| $(3b)$ | $t_3 = t_5$ |
| $(3c)$ | $t_3 = t_6$ |
| $(!4)$ | $\texttt{bool} = t_x$ |
| $(5)$ | $t_5 = t_p$ |
| $(6)$ | $t_5 = t_q$ |
| $(!7a)$ | $t_7 = \texttt{int}$ |
| $(!7b)$ | $t_8 = \texttt{int}$ |
| $(!7c)$ | $t_9 = \texttt{int}$ |
| $(9)$ | $t_{10} = t_{11} \to \texttt{int}$ |
| $(10)$ | $t_{10} = t_f$ |
| $(11)$ | $t_{11} = t_r$ |

Now, we could continue by processing the var/nonvar (2) and (9). But because the nonvar sides are a bit complex, it turns out to be easier to handle them later. Instead we will now process all the var/var equations. It doesn't matter which variable we keep in each case; for uniformity, we'll keep the type variables corresponding to (program) variables in preference to those corresponding to AST nodes. Doing this for $(1a), (10), (11)$ leads to:

(!1$a$) $t_f = t_2$
(!1$b$) $t_1 = \texttt{int}$
(2) $t_f = \texttt{bool} \to t_3$
(!3$a$) $t_4 = \texttt{bool}$
(3$b$) $t_3 = t_5$
(3$c$) $t_3 = t_6$
(!4) $\texttt{bool} = t_x$
(5) $t_5 = t_p$
(6) $t_5 = t_q$
(!7$a$) $t_7 = \texttt{int}$
(!7$b$) $t_8 = \texttt{int}$
(!7$c$) $t_9 = \texttt{int}$
(9) $t_f = t_r \to \texttt{int}$
(!10) $t_{10} = t_f$
(!11) $t_{11} = t_r$

Processing $(5, 3b, 3c)$ in that order gives the following (where the substitution from $(3c)$ has no effect):

(!1$a$) $t_f = t_2$
(!1$b$) $t_1 = \texttt{int}$
(2) $t_f = \texttt{bool} \to t_p$
(!3$a$) $t_4 = \texttt{bool}$
(!3$b$) $t_3 = t_p$
(!3$c$) $t_p = t_6$
(!4) $\texttt{bool} = t_x$
(!5) $t_5 = t_p$
(6) $t_p = t_q$
(!7$a$) $t_7 = \texttt{int}$
(!7$b$) $t_8 = \texttt{int}$
(!7$c$) $t_9 = \texttt{int}$
(9) $t_f = t_r \to \texttt{int}$
(!10) $t_{10} = t_f$
(!11) $t_{11} = t_r$

We choose arbitrarily to keep $t_q$ when processing (6), leading to

$(!1a)\quad t_f = t_2$
$(!1b)\quad t_1 = \texttt{int}$
$(2)\quad t_f = \texttt{bool} \to t_q$
$(!3a)\quad t_4 = \texttt{bool}$
$(!3b)\quad t_3 = t_q$
$(!3c)\quad t_q = t_6$
$(!4)\quad \texttt{bool} = t_x$
$(!5)\quad t_5 = t_q$
$(!6)\quad t_p = t_q$
$(!7a)\quad t_7 = \texttt{int}$
$(!7b)\quad t_8 = \texttt{int}$
$(!7c)\quad t_9 = \texttt{int}$
$(9)\quad t_f = t_r \to \texttt{int}$
$(!10)\quad t_{10} = t_f$
$(!11)\quad t_{11} = t_r$

Finally, we are back to our remaining var/nonvar equations. Arbitrarily choosing to process (2), we substitute as usual, ending up with:

$(!1a)\quad \texttt{bool} \to t_q = t_2$
$(!1b)\quad t_1 = \texttt{int}$
$(!2)\quad t_f = \texttt{bool} \to t_q$
$(!3a)\quad t_4 = \texttt{bool}$
$(!3b)\quad t_3 = t_q$
$(!3c)\quad t_q = t_6$
$(!4)\quad \texttt{bool} = t_x$
$(!5)\quad t_5 = t_q$
$(!6)\quad t_p = t_q$
$(!7a)\quad t_7 = \texttt{int}$
$(!7b)\quad t_8 = \texttt{int}$
$(!7c)\quad t_9 = \texttt{int}$
$(9)\quad \texttt{bool} \to t_q = t_r \to \texttt{int}$
$(!10)\quad t_{10} = \texttt{bool} \to t_q$
$(!11)\quad t_{11} = t_r$

This is the most interesting step. We are left with a nonvar/nonvar equation with an arrow type on each side. To process this, we must *replace* it with two new equations, equating the domain and range types of the arrows, thus:

$$
\begin{array}{rl}
(!1a) & \texttt{bool} \to t_q = t_2 \\
(!1b) & t_1 = \texttt{int} \\
(!2) & t_f = \texttt{bool} \to t_q \\
(!3a) & t_4 = \texttt{bool} \\
(!3b) & t_3 = t_q \\
(!3c) & t_q = t_6 \\
(!4) & \texttt{bool} = t_x \\
(!5) & t_5 = t_q \\
(!6) & t_p = t_q \\
(!7a) & t_7 = \texttt{int} \\
(!7b) & t_8 = \texttt{int} \\
(!7c) & t_9 = \texttt{int} \\
(9a) & \texttt{bool} = t_r \\
(9b) & t_q = \texttt{int} \\
(!10) & t_{10} = \texttt{bool} \to t_q \\
(!11) & t_{11} = t_r
\end{array}
$$

After processing these in the usual way, we have no more unprocessed equations, and we have reached a solution, which we can just read off from the processed equations, which each now equate a variable with a ground term.

$$
\begin{array}{rl}
(!1a) & \texttt{bool} \to \texttt{int} = t_2 \\
(!1b) & t_1 = \texttt{int} \\
(!2) & t_f = \texttt{bool} \to \texttt{int} \\
(!3a) & t_4 = \texttt{bool} \\
(!3b) & t_3 = \texttt{int} \\
(!3c) & \texttt{int} = t_6 \\
(!4) & \texttt{bool} = t_x \\
(!5) & t_5 = \texttt{int} \\
(!6) & t_p = \texttt{int} \\
(!7a) & t_7 = \texttt{int} \\
(!7b) & t_8 = \texttt{int} \\
(!7c) & t_9 = \texttt{int} \\
(!9a) & \texttt{bool} = t_r \\
(!9b) & t_q = \texttt{int} \\
(!10) & t_{10} = \texttt{bool} \to \texttt{int} \\
(!11) & t_{11} = \texttt{bool}
\end{array}
$$