“How to Solve Recursive Equations”
(of monotone functions over lattices)

Claus Brabrand
ITU, IT University of Copenhagen
(April 3, 2008)

Let’s solve the rec. equations:

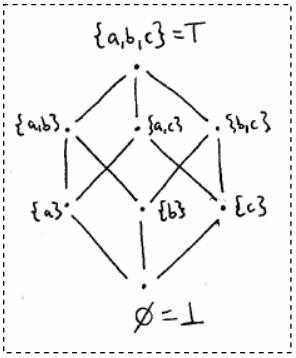
$$X = \{a,b\} \cup Y$$
$$Y = X \cap \{b\}$$
$$Z = Z \cup Y$$

…over the (power-)lattice:

“ $P(\{a,b,c\})$ ”

(which is also known as:

“ $2^{\{a,b,c\}}$ ”)



1. Transform Equations

First, we bring the equations onto the form we know how to handle; i.e.:

$$x_1 = f_1(x_1, x_2, \ldots, x_n)$$
$$x_2 = f_2(x_1, x_2, \ldots, x_n)$$
$$\ldots$$
$$x_n = f_n(x_1, x_2, \ldots, x_n)$$

In our case, this becomes:

$$X = f(X, Y, Z)$$
$$Y = g(X, Y, Z)$$
$$Z = h(X, Y, Z)$$

…where:

$$f(X,Y,Z) = \{a,b\} \cup Y$$
$$g(X,Y,Z) = X \cap \{b\}$$
$$h(X,Y,Z) = Z \cup Y$$

(All we really did was to make the functions involved explicit.)

We now see that we have three functions we need to show are monotone (in all arguments). But before we do that, let’s first simplify the equations by removing arguments the functions “don’t use” (i.e., arguments in which the functions are constant; or, equivalently, arguments on which the resulting value of the function do not depend). We then get (I arbitrarily chose to capitalize the names of the “new” functions which take a bit fewer arguments):

$$X = F(Y)$$
$$Y = G(X)$$
$$Z = H(Y, Z)$$

…where:

$$F(Y) = \{a,b\} \cup Y$$
$$G(X) = X \cap \{b\}$$
$$H(Y,Z) = Z \cup Y$$

2. Check for Monotonicity

Now we need to check monotonicity of the functions (in all arguments):

I'll just do "F". (The others are similar, although "H" needs to be checked for monotonicity in *both arguments*.)

Monotonicity means that I need to check that:

$$\forall\ Y,Y' \in P(\{a,b,c\}):\quad Y \subseteq Y' \quad\Rightarrow\quad F(Y) \subseteq F(Y')$$

…i.e., every time the argument to "F" gets bigger, the result of the function applied to the argument also gets bigger.

For our definition of "F" this means that we need to check that:

$$\forall\ Y,Y' \in P(\{a,b,c\}):\quad Y \subseteq Y' \quad\Rightarrow\quad \{a,b\} \cup Y \subseteq \{a,b\} \cup Y'$$

This clearly holds for any two sets Y and Y' (but a real proof would involve checking all eight ($= 2^{|\{a,b,c\}|}$) elements of the power-lattice in place of Y as well as Y'; i.e. 64 combinations!).

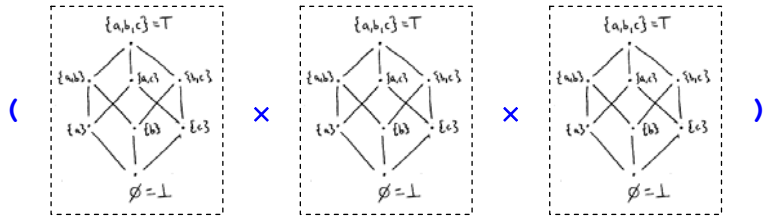
(When you get a bit more "fluent" in this extra-terrestrial math, you probably don't need to introduce explicit functions as we did here, but just think of "$\{a,b\} \cup Y$" as an implicitly defined function. I just introduced the functions "f", "g", and "h"; subsequently, "F", "G", and "H", in order to be explicit about *what* is checked for monotonicity.)

Now that we know that our functions "F", "G", and "H" are monotone (in all arguments) and that our partial-order is a lattice (of finite height), we can use the *Fixed-Point Theorem* to solve our recursive equations (in finite time).

3. Transform into 1 "big" lattice and 1 "big" function

We do this by transforming the problem so that we only have 1 "big" monotone function on 1 "big" lattice:

Lattice: In our case, we need 3 copies of the lattice (one for the value of X, one for Y, and for Z); we will thus be operating on the following "big" lattice:



Function: In our case, we need a function that operates on 3-tuples which is done by the following "big" function (let's call it "T"):

$$T((X,Y,Z)) = (F(Y),\ G(X),\ H(Y,Z))$$

…which, for instance, turns argument $(\emptyset,\ \{b\},\ \{a,b,c\})$ into

$$T((\emptyset,\ \{b\},\ \{a,b,c\})) = (F(\{b\}),\ G(\emptyset),\ H(\{b\},\{a,b,c\})) = (\{a,b\},\ \emptyset,\ \{a,b,c\})$$

…which is apparently *not* a fixed-point (since $(\emptyset,\ \{b\},\ \{a,b,c\})$ isn't equal to $(\{a,b\},\ \emptyset,\ \{a,b,c\})$).

4. Solve Equations using the *Fixed-Point Theorem*

The Fixed-Point Theorem now says that we can solve the equations (i.e., find the *least* fixed point) via simple iteration, by computing:

$$fix(T) = \cup_{i \geq 0}\ T^i(\_|\_)$$

…which boils down to computing $T^i(\_|\_)$ for increasing values of i:

$$(\emptyset,\emptyset,\emptyset) \subseteq T((\emptyset,\emptyset,\emptyset)) \subseteq T(T((\emptyset,\emptyset,\emptyset))) \subseteq T(T(T((\emptyset,\emptyset,\emptyset)))) \subseteq \ldots$$

…until nothing changes (i.e., we get two consequtive elements of the big lattice that are equal). In our case the iteration looks like:

#iterations	"$T^i(\_|\_)$" (i.e., i^{th} iteration)
1	$T^1(\_|\_) = T((\emptyset,\emptyset,\emptyset)) = (\{a,b\},\emptyset,\emptyset)$
2	$T^2(\_|\_) = T((\{a,b\},\emptyset,\emptyset)) = (\{a,b\},\{b\},\emptyset)$
3	$T^3(\_|\_) = T((\{a,b\},\{b\},\emptyset)) = (\{a,b\},\{b\},\{b\})$
4	$T^4(\_|\_) = T((\{a,b\},\{b\},\{b\})) = (\{a,b\},\{b\},\{b\})$

i.e., we find *the (unique!) least fixed-point* in just four iterations, since (as evident in the table): $T^3(\_|\_) = T^4(\_|\_) = (\{a,b\},\{b\},\{b\}) = T((\{a,b\},\{b\},\{b\}))$ which is then the least solution to the original equations; i.e.:

The *least* solution
$X = \{a,b\}$ and $Y = \{b\}$ and $Z = \{b\}$

Alternatively, you "guess" the solution and check it, but then you also have to reason about it being the *smallest* solution (i.e., the *least* fixed point). :-)

Other Solutions

Note that this set of equations has other (bigger) solutions; e.g.:

$$X = \{a,b\} \text{ and } Y = \{b\} \text{ and } Z = \{b,c\}$$

…and (even bigger solution)…:

$$X = \{a,b\} \text{ and } Y = \{b\} \text{ and } Z = \{a,b,c\}$$

.