Logical Constraints in Integer Programs

November 3, 2007

All throughout this document I'm going to use three simple statements, which I'm going to name X, Y, and Z, to illustrate the main concepts:

1. $X =$ "The sun is shining."
2. $Y =$ "I'm going to fail this class."
3. $Z =$ "I'm happy."

Adding logical constraints to our integer program means that we specify additional linear equalities or inequalities that express, in an algebraic manner, that certain ("logical") statements are true. In order to find such constraints we need to have *binary variables* which indicate whether the statements that appear in our constraints are true or false. Such a logical constraint could be something like "When the sun is shining, I'm happy", or "If I'm happy, then either the sun is shining or I'm not going to fail this class".

First we need to associate variables to all the simple statements that our constraints consist of. Clearly, in all the above examples these statements are X, Y, and Z; let us associate the variables x, y, and z to them. What this *associating indicator variables* precisely means is that, for example, the variable x is defined as

$$x = \begin{cases} 1 & \text{if the sun is shining} \\ 0 & \text{if it is not,} \end{cases}$$

or, to put it more generally,

$$x = \begin{cases} 1 & \text{if } X \text{ is true} \\ 0 & \text{if } X \text{ is false,} \end{cases}$$

You can guess now how y and z are defined.

You can see, and this is an important convention, that the value 1 is associated to TRUE statements, and the value 0 is associated to FALSE statements.

At this point, it would be trivial to add constraints like "the sun is shining". All we would have to do is to say $x = 1$. Similarly, "I'm not happy" translates to $z = 0$.

So far so good, but how can we put more complicated expressions in algebraic form? There are two key points here: first we will see how to translate slightly more complicated statements like "the sun is shining and I'm happy", and then we will see how every really complicated case reduces to the slightly complicated cases.

For now let us forget about using linear constraints. I will allow now to multiply variables (always forbidden when dealing with linear stuff!), and at the very end we will see that we can always *linearize* these products.

Let us see now how to define variables for expressions which involve AND and OR. If we want a variable w which is defined as

$$w = \begin{cases} 1 & \text{if } X \text{ and } Y \text{ are true} \\ 0 & \text{otherwise} \end{cases},$$

then we can simply put this as

$$w = xy.$$

Indeed, w will be 1 if and only if $x = y = 1$, that is, if both X and Y are true. In every other case $w = 0$, which is precisely what we wanted. So to write the algebra for "X and Y are both true", we simply have to add the constraint

$$xy = 1$$

How about OR? It is only slightly more complicated:

$$w = \begin{cases} 1 & \text{if } X \text{ or } Y \text{ is true} \\ 0 & \text{otherwise} \end{cases},$$

is the same as saying

$$w = x + y - xy.$$

Again, it is not hard to check that if at least one of X and Y is true, so $x = 1$ or $y = 1$, then $w = 1$ as expected, but if none of them is true, so $x = y = 0$, then $w = 0$. This is exactly what we wanted from w. Yet, whenever we have to say that "X or Y is true", we can do it in a simpler way: instead of writing $x + y - xy = 1$, we can simply write

$$x + y \geq 1.$$

It should be clear at this point, why this is correct. Notice the difference: the second, simple formulation is only to say that X or Y is true". To introduce a new variable w for the statement $W = X$ or Y is true we have to use $w = x + y - xy$.

There is a third simple case we can quickly discuss. What if we need a variable to express the statement "I'm NOT happy"? This can be written as

$$w = \begin{cases} 1 & \text{if } Z \text{ is false} \\ 0 & \text{if } Z \text{ is true} \end{cases},$$

which clearly translates to

$$w = 1 - z.$$

If we had to list and memorize the algebraic expression for every logical function possible, we would need to write a pretty long list. But the good news is: *everything can be expressed using only these three building blocks: AND, OR, and NOT.* This doesn't mean that everything always should be expressed only these. Many times there are more convenient ways. In particular, I recommend that you remember this simple one: "either X or Y" translates to

$$x + y = 1.$$

I'm going to refer to this as "EITHER-OR". (It is also frequently called XOR, which stands for "exclusive or").

Example. So what's the algebra for "I'm happy and I'm going to fail the class and the sun is shining"? This is just two ANDs, so we need to multiply everything to get: $xyz = 1$. How about "I'm not happy, but the sun is shining, and yet, I'm not going to fail the class"? Notice that things like "but" and "yet" have nothing to do with logic, so we can strip down this colorful sentence to get "I'm NOT happy, AND the sun is shining, AND I'm NOT going to fail the class", which will be

$$(1 - z)x(1 - y) = 1.$$

As I noted before, *everything can be expressed using only these three building blocks: AND, OR, and NOT.* So if whatever constraint like "something-something is TRUE" you have, you can always put it as "$\cdots = 1$", where "$\ldots$" is the (nonlinear!) algebraic expression for "something-something". But as we have seen in the case of OR, there can be simpler solutions using inequalities. Recall that "X or Y" was simply put as $x + y \geq 1$. A similar, very important example is the following

Example. "If the sun is shining, then I'm happy". This translates to "the sun is not shining OR (the sun is shining AND I'm happy)", which we could, in principle, write as $(1 - x) + (xz) - (1 - x)xz = 1$. (See how I got this one? I replaced NOT by "1-", AND by multiplication and OR by the expression $x + y - xy$.) This looks awful, couldn't we write it simpler? Well, we can, just notice that the last term, $(1 - x)xz$ is always zero, because either $1 - x = 0$ or $x = 0$. So this is really the same as simply $1 - x + xz = 1$, or $x + xz = 0$. But there is an even simpler way, we have already used this in class:

$$x \leq z.$$

Before we turn to linearizing the expressions involving AND, let us see a more complicated example.

Example. "If I'm happy, then the reason is either that the sun is shining or that I'm not going to fail the class, but it can be both, too." This (again, common sense!) simplifies to "IF I'm happy, THEN (the sun is shining OR I'm NOT going to fail the class)". So how do we do this? The whole phrase is an implication, so if I denote by W the statement "the sun is shining OR I'm NOT going to fail the class", then the whole phrase is just $z \leq w$. But as we have seen before, that the newly introduced variable can be written as

$w = x + (1 - y) - x(1 - y)$. (I used the rules for OR and for NOT.) Of course, the right hand side could be simplified, but let's not worry about that now. So we obtained the inequality

$$z \leq x + (1 - y) - x(1 - y).$$

Finally, let's see what can we do to get rid of all the nonlinear expressions. I'm going to use the last example to illustrate the idea.

First, let us expand all the parentheses:

$$z \leq x + 1 - y - x + xy,$$

and then do the cancelations:

$$z \leq 1 - y + xy.$$

Apparently, there is only one nonlinear term, xy. Let us *introduce a new variable for it*, say, w, and then write our constraints as

$$\begin{aligned} z &\leq 1 - y + w \\ w &= xy \end{aligned}$$

Now it looks like we haven't gained anything, we still have that xy lying around. But this last constraint is equivalent to the following two inequalities:

$$2w \leq x + y \leq w + 1.$$

(Check, why??)

Notice that this one last trick is always enough, in every possible example. After we expand all the parentheses, we end up with a bunch of terms, which are either linear, or products of variables. We never need to have powers: things like x^2 or x^3 always simplify to x, because x is zero or one, and every power of zero and one are just themselves: $x = x^2 = x^3 = \ldots$. So the only nonlinear terms we can have are products of different variables, and this is what we tackle by introducing a new variable, and replacing the $w = xy$-type equalities by two linear inequalities each.