We've added a domino via $DRSK$ to the number tableau and the dual number tableau. This has changed the shape of the number tableau by adding *position* to its shape, and the dual number tableau by adding *dualPosition* to its shape. (TODO, prove that no other cases occur.)

The cases depend on the grid sub positions of *position* and *dualPosition*, and on whether they are horizontal or vertical. Sometimes there are further requirements for a case. Most (all?) of the cases are symmetric. That is to say, they may require that both positions satisfy the same conditions. Alternatively, they may require that one of the tableaux satisfy some condition. In the latter case, we'll assume for simplicity that the the tableau which satisfies it is the $+,-$ tableau.

For most cases, we incorporate *position* and *dualPosition* into the sign tableaux, and then add the larger number with some sign to one of the two sign tableaux, using `addNumberSign()`. In those cases, we'll choose the sign to place in either the *position* or *dualPosition* domino to be compatible with signs which are already present in the tableau. After that, the opposite sign will be passed to `addNumberSign()`. If instead we're in one of the exceptional cases, we'll indicate that.

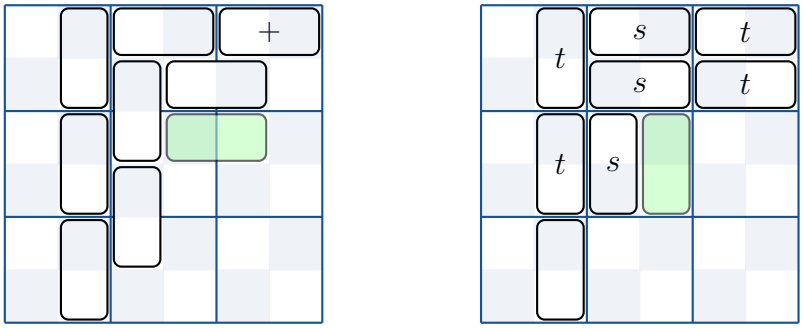
Write *gpos* (respectively *dgpos*) for the grid sub position of *position* (respectively *dualPosition*). Let x, y be the coordinates of the first square of *position*.

- Here $gpos = Y$ and *position* is horizontal. The dual domino is vertical, starting in square y, x. (TODO, prove.) There are two main cases.
  - ▷ Here the next row has the same length. If we're not in the first column, then to the left is either a box or two nested cycles. So, we have both signs available to the left. We'll put a box here with both dominoes, and, if necessary, move it up.

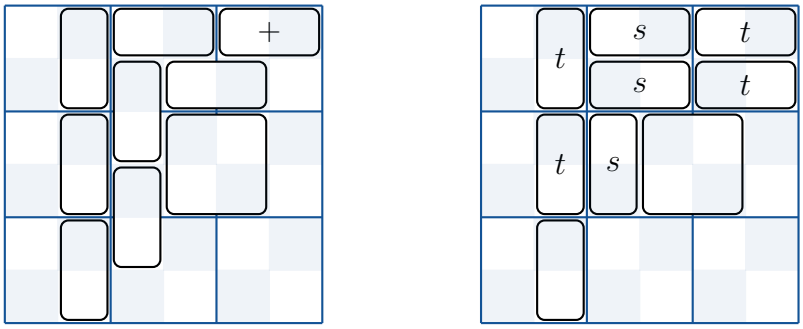
    A simple example starts with empty tableaux.

    goes to

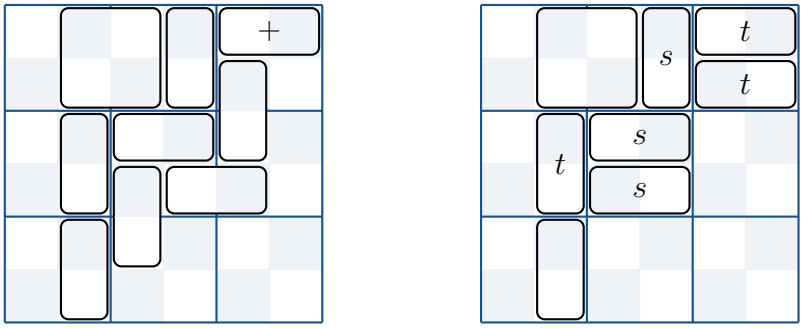
Here is a more complicated example.

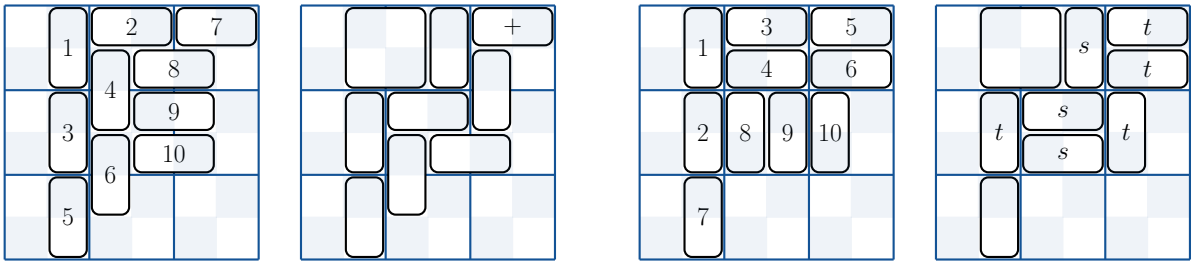


goes to



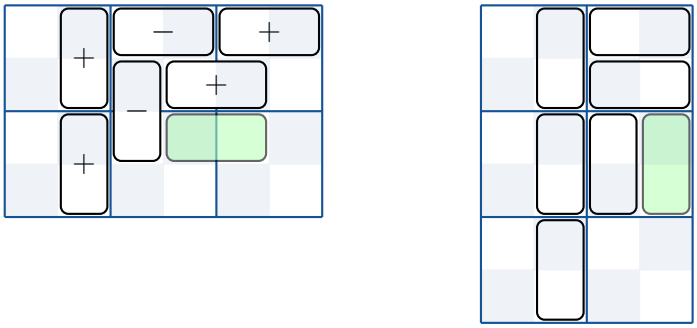
goes to



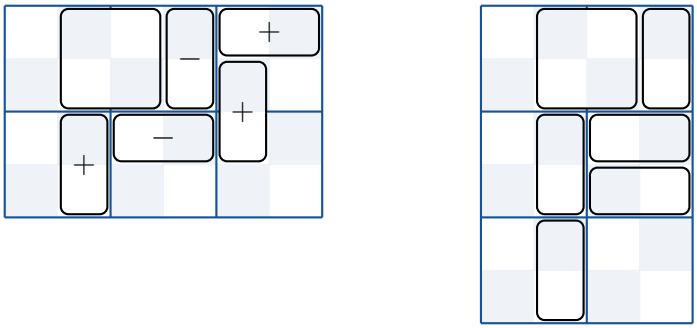


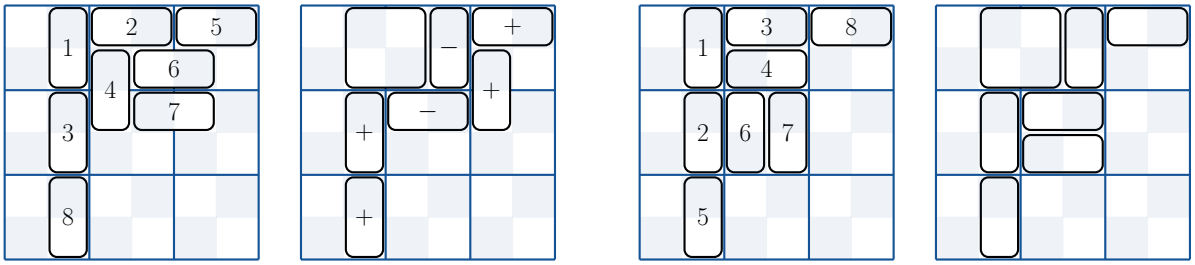
▷ Here the next row has a shorter length. We'll use the new domino to contract an unboxed cycle.

Basically, there are two cases for this insertion. On the one hand, there may be a sign in the vertical domino to the left. In that case, we give the new domino the opposite sign. This is compatible with the signs already in the cycle. So, we just add the domino, giving it the opposite sign to the one on the left, box things up, and then call `addNumberSign()` with the same sign as on the left.



goes to

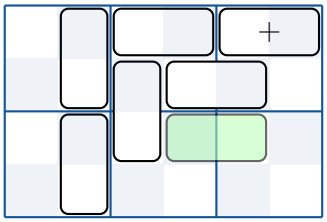


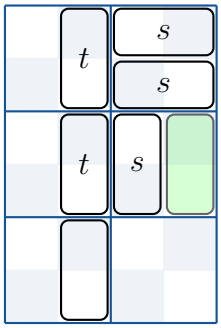


On the other hand, it may be that the domino to the left is blank. Now, note, the domino to the left is a cycle bottom. So, it follows that even if the left domino is horizontal, there is effectively no sign at that level. The only possible signed domino in that row would be a vertical domino which we can move down if necessary by calling `makeSpaceFor()`, which we do. (Any other sign would have already moved to the left domino, since any sign can go in a cycle bottom.) In addition, once we add a domino here, with either sign, the next domino will be able to add at the next row, and both signs will be blanked in that process, leaving this cycle with the same signs which it started with. So, that's what we expect to see, and what we see in the example below.

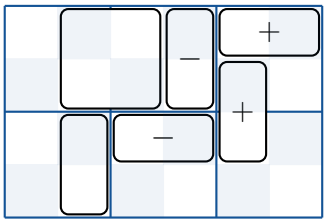
So, for this step, if there is no sign in the left domino, we just choose the sign to add here for convenience, so that putting it in the cycle bottom doesn't move the cycle. So, we choose a compatible sign. To do that, we look at the top domino. If it has a sign, we choose its sign. If not, we look at the cycle top. If it has a sign, we use the function `getExpectedSign()` to choose a sign for the newly-added domino. If not, we choose a sign arbitrarily. In any case, once we've chosen the sign, we call `putSignInCycleBottom()`.

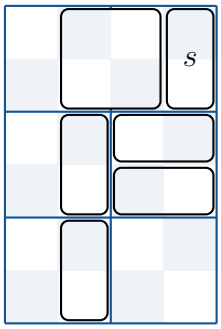
Here is an example.

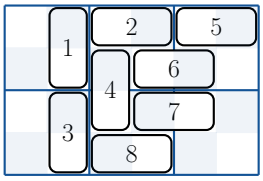


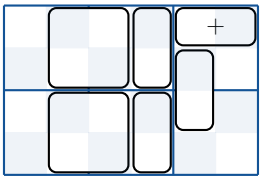


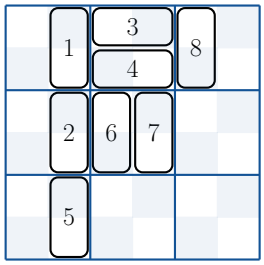
goes to

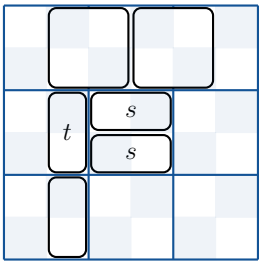




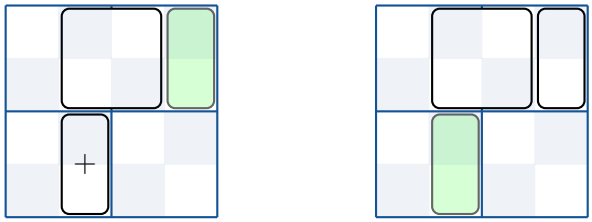




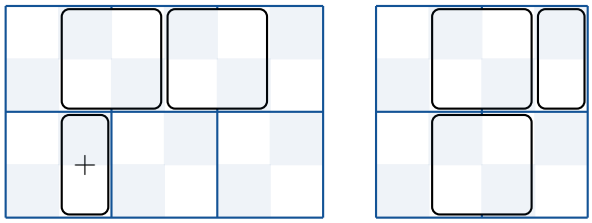


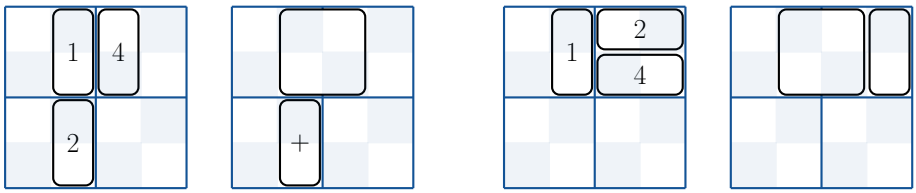


- Here *gpos* and *dgpos* are both Y, and both *position* and *dualPosition* are vertical. This is an exceptional case. We make a box here, with both dominoes. Note, it's a little pecular that both positions come out vertical. This may need some discussion. In the example, one can see the issue that the dual number tableau is not just transposed but also cycled through. Probably, something needs proving here.

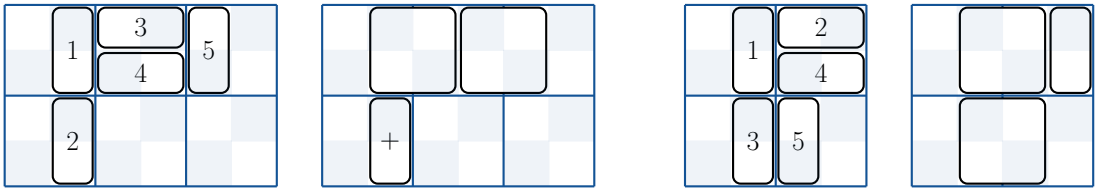


goes to



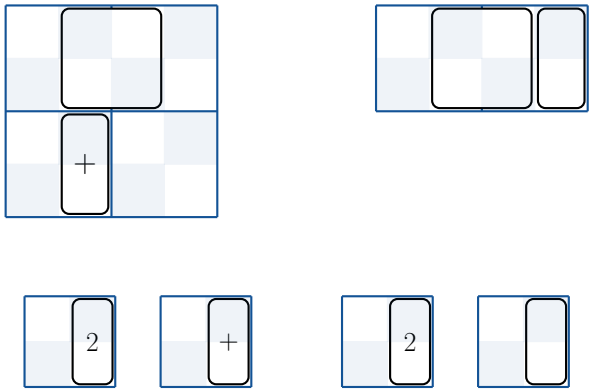


goes to

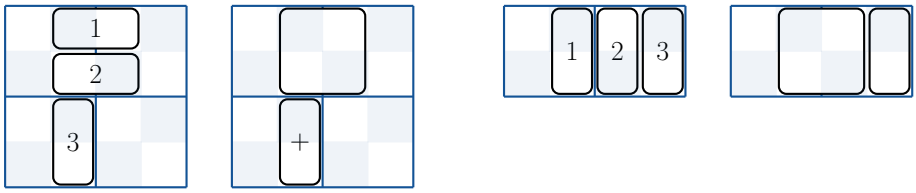


- Here *gpos* and *dgpos* are both X, and both *position* and *dualPosition* are vertical. As in the previous case, this has to do with being cycled through. On one side the new position will be next to a vertical domino with a sign in it (which we will assume to be plus), on the other side enxt to a blank domino. We will box up the new positions, and add a + sign below the former + sign domino.

goes to



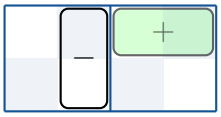
goes to

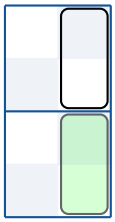


- Here $gpos = X$ and *position* is horizontal. There are a number of cases. We split on how the cycle structures of the tableaux are changed by the addition of the dominoes. (Note, I wrote about using `makeSpaceFor()`, but actually, I didn't change the code yet to use that function.)

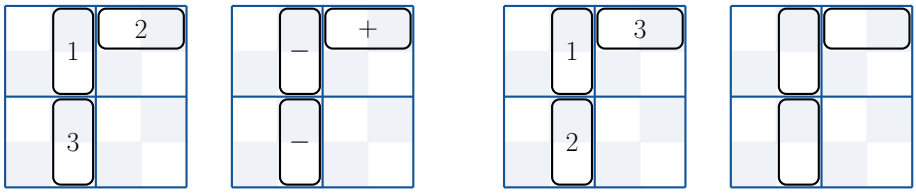
  ▷ Here the added domino extends a Type I cycle. If there's a sign in the domino directly to the left, we put the opposite sign into the newly-added domino; otherwise we choose an arbitrary sign. In the case where we choose an arbitrary sign, that choice will be blanked once the next domino is added at the row after. For now, we make sure we can place the arbitrary sign here by calling `makeSpaceFor()`. Note, there is no need to avoid the effort of moving a sign down by making a sign choice which is compatible with whatever sign is at the end of this row. If there is a sign at the end of the row, it would have to move down anyway when we come to add the second domino of the pair.

  Here are three examples.

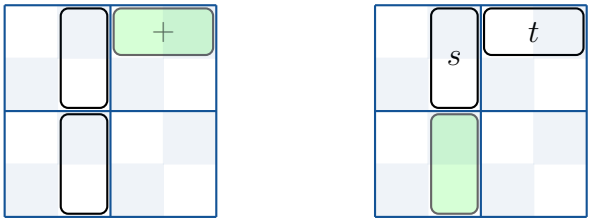


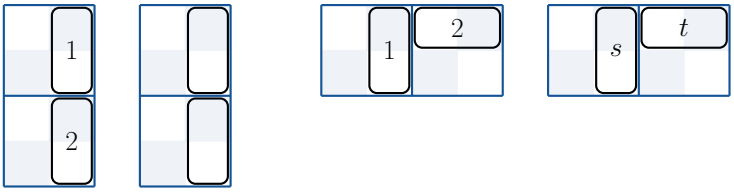


goes to

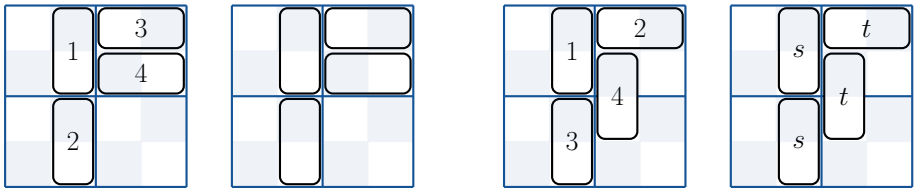


Next example:

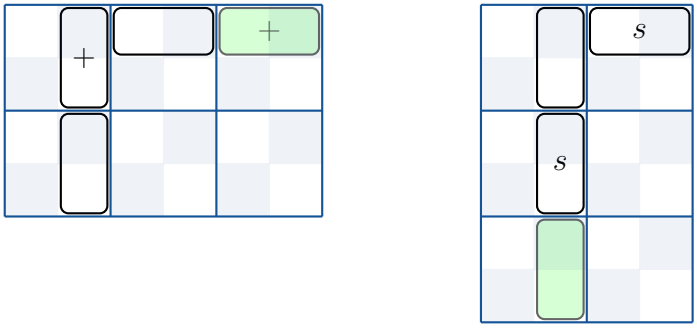




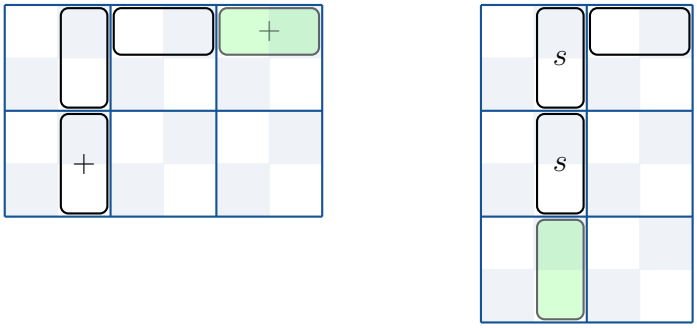
goes to

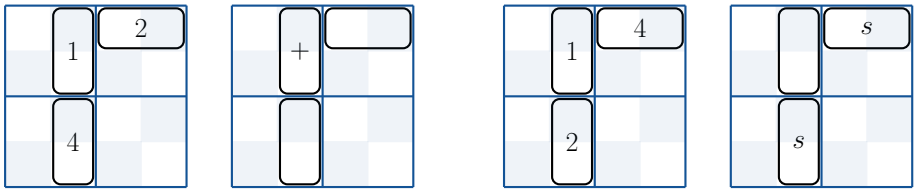


Third example:

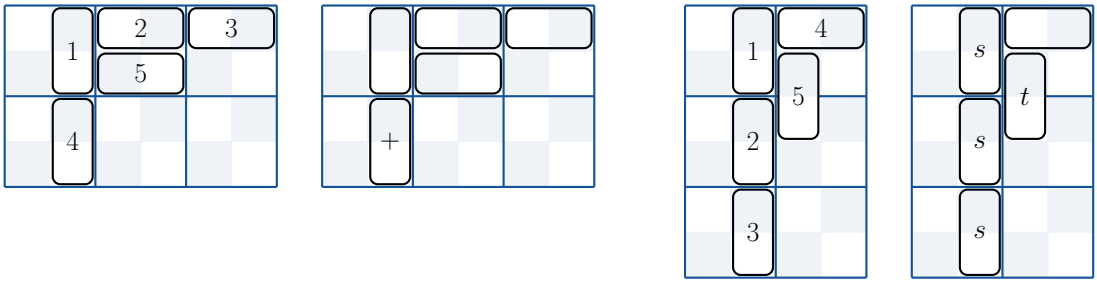


goes to (via `makeSpaceFor()`)

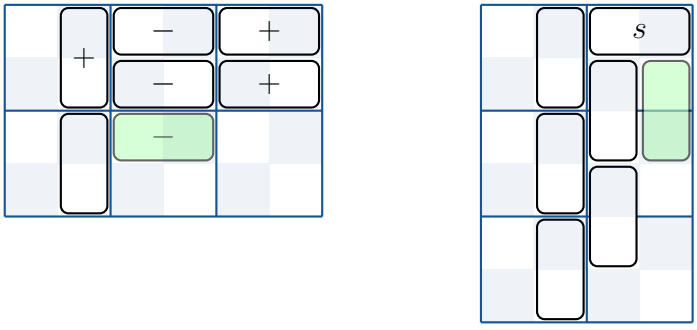




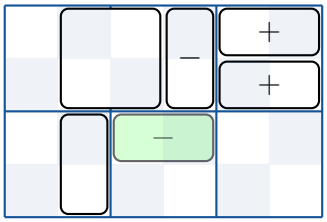
goes to

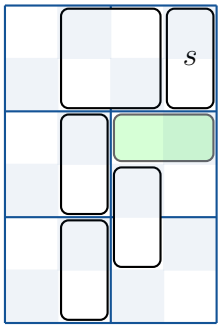


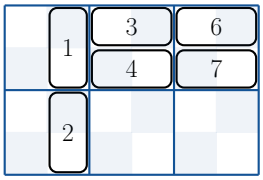
▷ Here we are contracting a boxed Type II cycle. Again, if there is a sign to the left, we take the opposite sign to it. Also, again, if there is blank to the left, we don't care that much which sign we put here, since the sign will be blanked when the second domino is added. However, after we place the new domino and make a box, the new domino will be attached to the domino two rows above it. So, if the domino two rows above has a sign, we'll choose the same sign for the domino which we are adding. (Note, the side domino is the cycle bottom, whereas the above domino is a paired domino in the cycle. So, if they both have signs, they will have opposite signs. So, in that case, the two prescriptions have the same result.) If neither has a sign, we make an arbitrary choice. As above, if the side domino is blank, we call `makeSpaceFor()` before adding the domino. In all the cases, after we add the dominoes, we box them up.

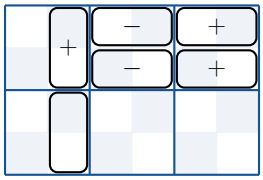


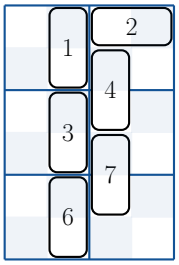
goes to

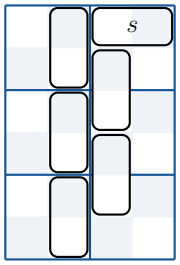




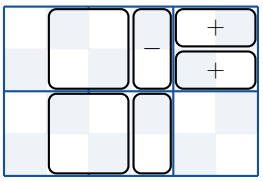


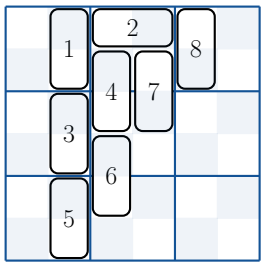


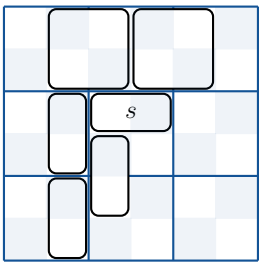




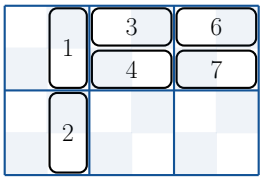
goes to

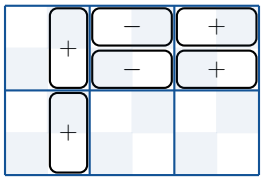


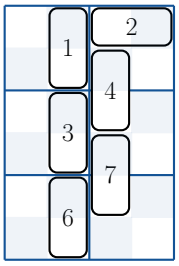


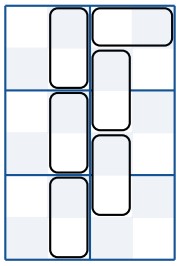


Also,

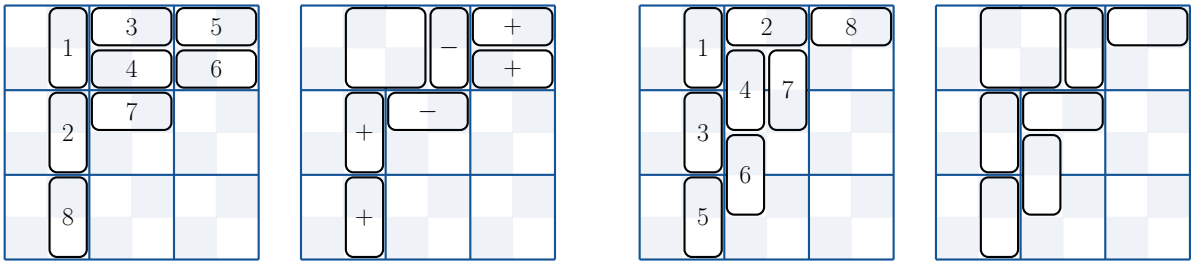








goes to



▷ Here we are closing a boxed Type II cycle (which therefore contains just one domino, the one directly above this one). So, we are interested in four dominoes to start with, namely the cycle top and bottom and the two paired dominoes. There are five possible configurations for how these four dominoes are filled with signs.

I think, again, we don't care that much what sign we put in if the side domino is blank, since that sign will be blanked when the next domino is added to the next row. However, we might need the tableau to be in a consistent state for the next domino to be added. Luckily, I have notes about that now.

Another issue is that, if the top domino is blank, and if the sign we are adding does not end up being blanked, then we are adding a sign to that column which may be incompatible with whatever is above the top domino. So we need to bring that sign down first.

Also, after adding a sign, we may be able to move it up in its column.

Let's look at the cases. In all the cases, after we add the dominoes, we box them up.

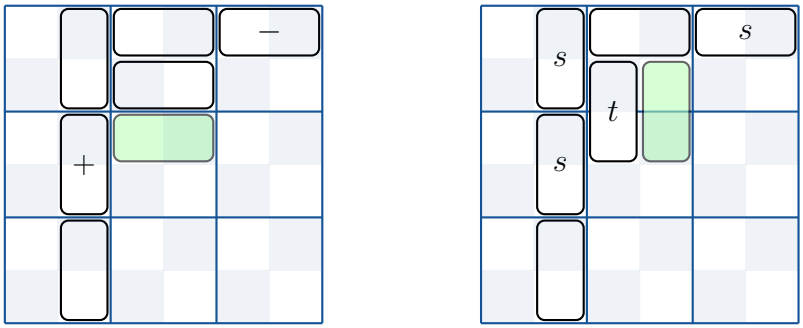
* If the side domino is blank, we can add a horizontal domino here with either sign. The sign will be blanked when the next domino is added. For convenience, if the top domino has a sign, or if any domino in its column has as sign, we choose that sign. Otherwise we choose a random sign. As above, if the side domino is blank, we call `makeSpaceFor()` before adding the domino. (TODO pictures.)
* Here the side domino is vertical and has a sign and there is a blank domino in its column. We need to move the blank domino up. We do this using `findRowToAddSignX()`, with the variation that we will not be moving through cycles in the number tableaux at this location. (The cycles are al-

ready closed.) However, if `findRowToAddSignX()` makes a shape change in the sign tableaux, we will add the signed domino in the dual sign tableau.

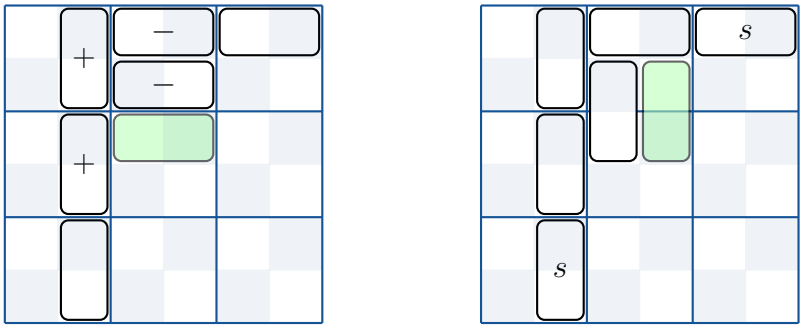
Basically, if there is a shape change, then we switch sides and do this situation, but on the other side. However, if we do that, then in one specific case, we could have another shape change and end up back on this side. Instead of doing that, we handle that as a special case, before calling `findRowToAddSignX()`. This will be the first case listed.

Let's do the cases.

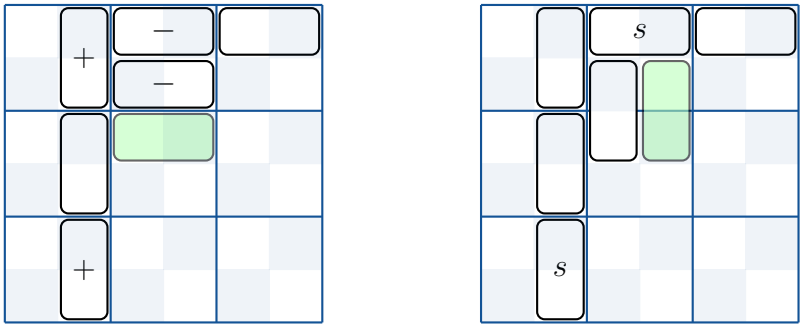
- ⋈ Here the side domino has a + (for convenience) sign, the top domino is blank, the top domino's row has a minus sign in it. On the dual side, the top corner domino has an s (for convenience), and there is an s sign in its row. Before calling `findRowToAddSignX()`, we pull the $-$ sign intto the top domino. (Note, if we didn't do this, we could instead proceed to the next case. But, then we would end up having two shape changes.)

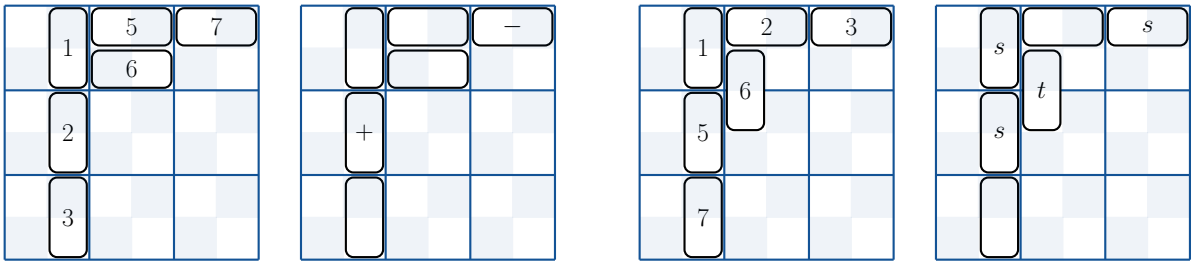


goes to

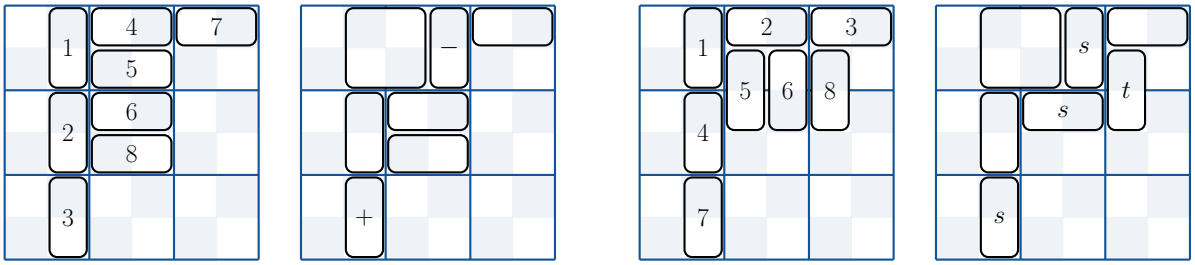


goes to

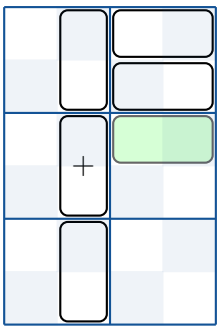


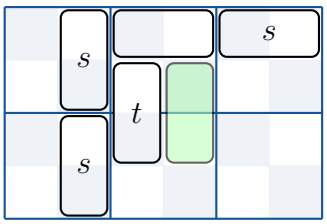


goes to

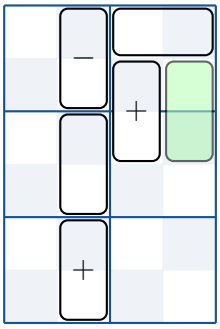


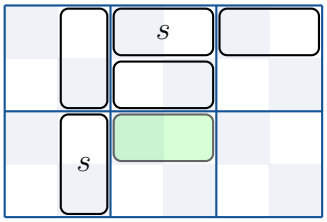
⋈ Here `findRowToAddSignX()` makes a shape change in the sign tableaux. So, basically, we are in the same situation as before, except on the other side. However, we know that the top domino has a sign. Our next step is to see if the side domino has a sign, and, if so, if we can bring up a blank domino. Note, this will not cause a shape change. We have eliminated that case in the previous case. Also note, this example ends up in the hard case.

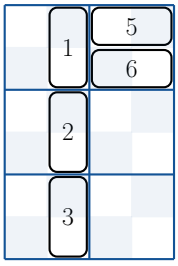


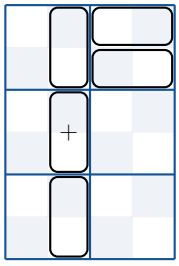


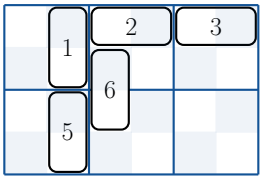
goes to

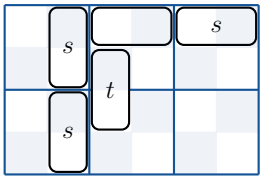




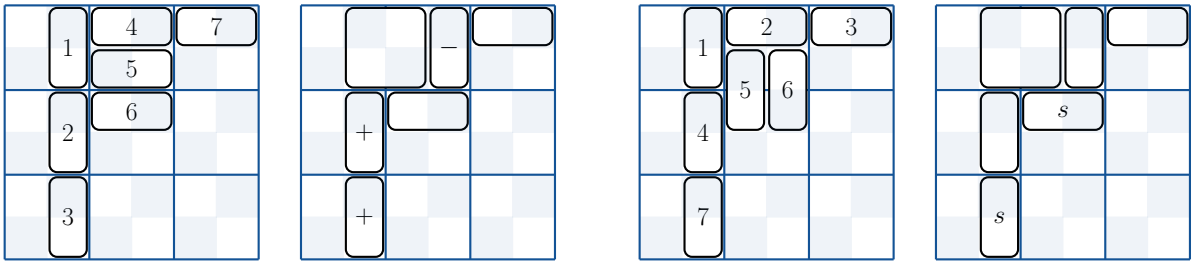








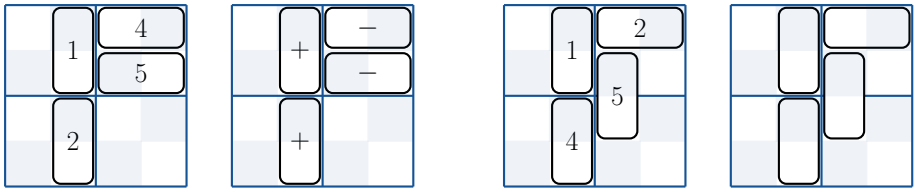
goes to



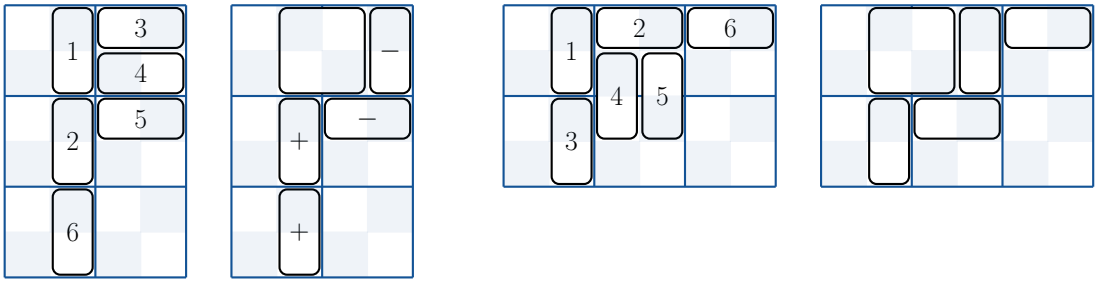
⋈ Otherwise, we can just proceed as if there was a blank in the side domino from the beginning.

∗ Here the side domino is vertical and has a sign (+ for convenience) and there are no blank signs in its column. So, basically, we just put a − sign in the newly-added domino. There are two cases where that causes problems. One is where the top domino also contains a + sign. The other is when the top domino is blank, and the column above it has a + sign. Let's see the base case and these special cases.

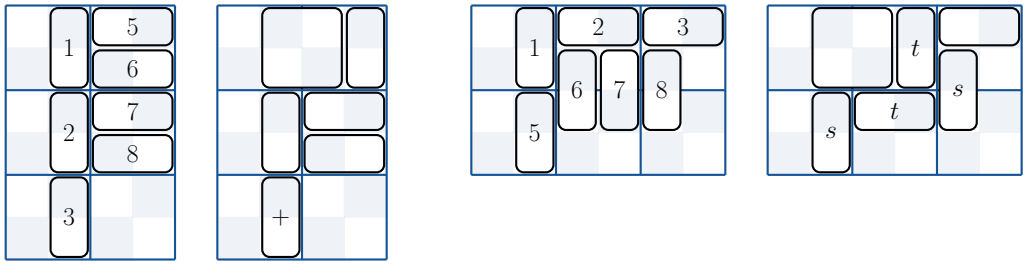
⋈ Here is an easy case:



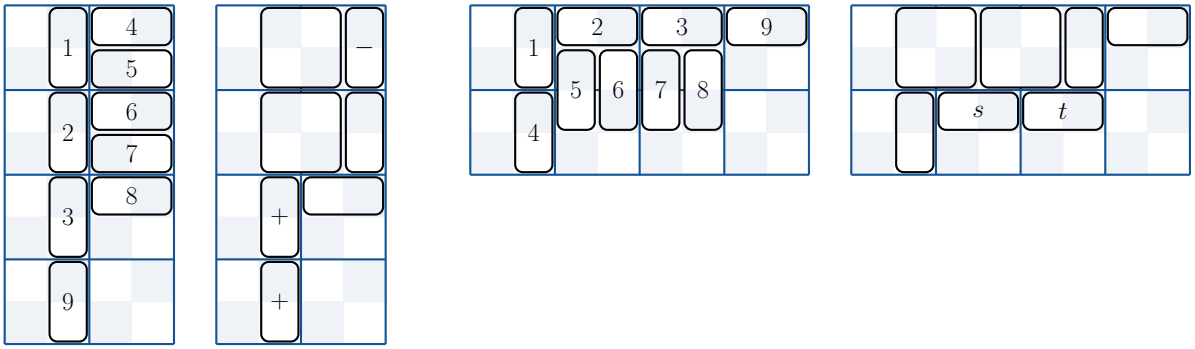
goes to



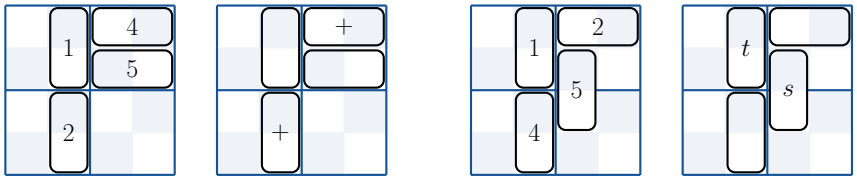
⋈ Here there is a blank domino in the column above. The newly-added domino can move up.

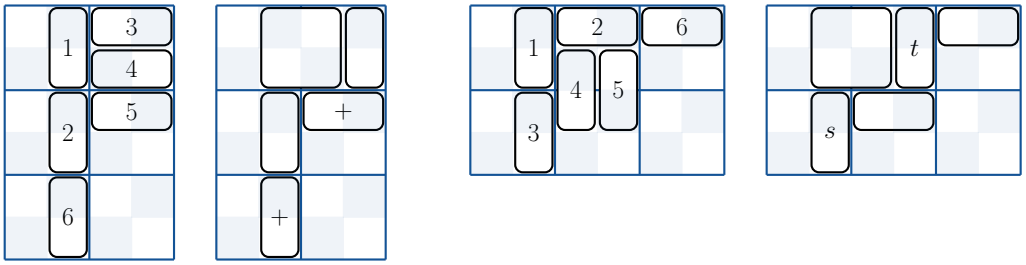


goes to



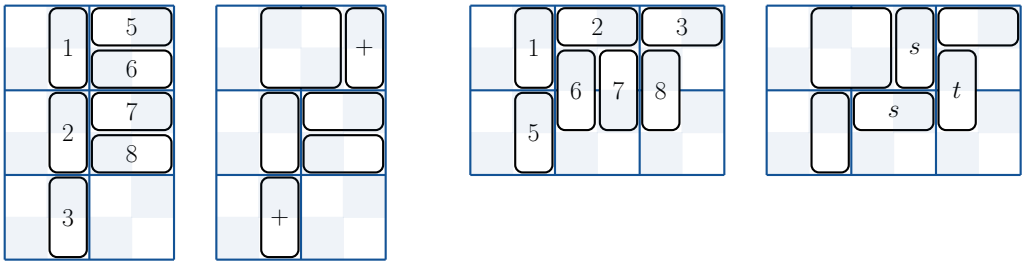
⋈ Here the top domino also contains a + sign. Here we basically undo what was done when two signs on the dual side were trapped by the two + signs. We free them and separate them. On the left, the − sign combines with one of the + signs, and they disappear, leaving only one + sign, in the middle. First, a basic example.



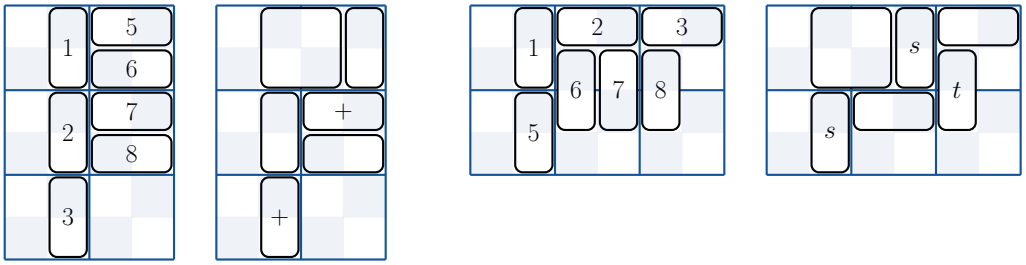


Note, the t is in a row of all blanks, by our assumption that the side column on the + side had all + signs. The easy issue is that we need to make space for the lower sign

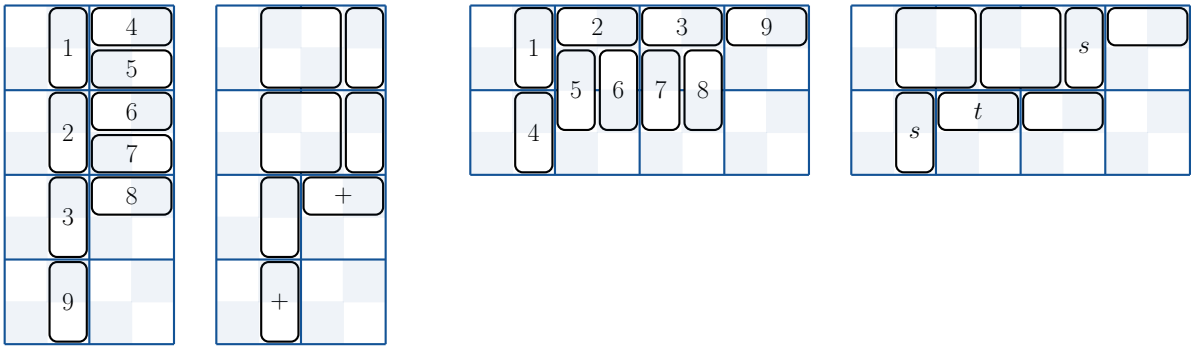
⋈ Here the top domino is blank, and the column above it has a + sign. We move it down first. Then we are in the previous case.



goes to



goes to



∗ Here the side domino is horizontal and has a + sign. In this configuration, there is a column above the top domino on the dual side. Basically, we proceed as before, except in one case, which we'll describe. In this case, the top domino also contains a + sign. Assume s is in the top corner domino. If there is a t in the column above the top domino, we can't proceed as usual. Instead, pull the t down first. Now the side domino has no sign, and we can put the − sign there, by swapping it with the blank corresponding to the t domino. Now the t domino is lower, so we need to call `makeSpaceFor()` for it.