

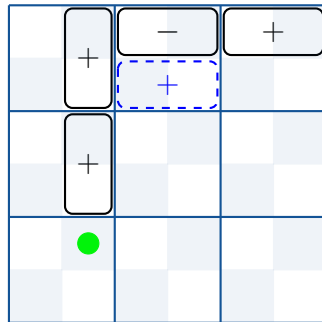
Next up is grid position Z . As before, we'll assume the sign which we are adding is $+$. We're looking to add at position (x, y) . We need to look at the domino which will be the pair of the new domino, that is, the domino occupying the square at position $(x-1, y-1)$. We'll call it the pair domino. The situation splits into whether the pair domino is horizontal or not. When the pair domino is vertical, we'll use the helper function `findRowToAddSignZ()`.

Let's look at `findRowToAddSignZ()` first. In this function, the pair domino is vertical.

- If the pair domino has the opposite sign, we're good to go.

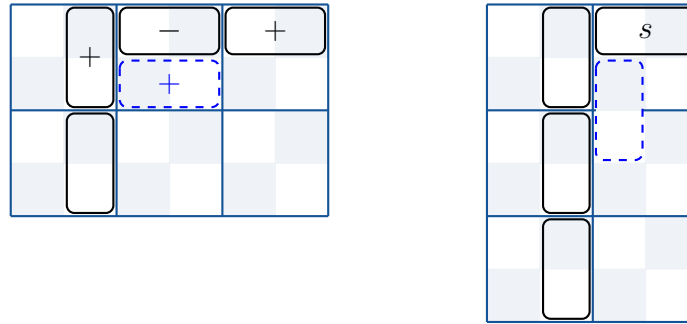


- If the pair domino has the same sign, we need to see if there is a blank domino in the same column. Concretely, we call the function `findSignInRowRight()` on the corresponding row in the dual sign tableau.
 - ▷ If `findSignInRowRight()` doesn't find a sign (or if we didn't bother to call the function because there's only one domino in the column), that means that our column is full of $+$ signs. We look next at the row below the end of this column.

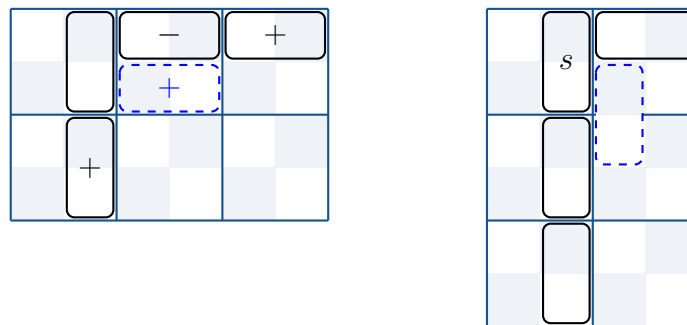


- ▷ If `findSignInRowRight()` finds a sign, we'll interchange the signs. We'll do the preparation and move the signs in this function. (Note, some of these cases can be complicated, possibly related to complicated cases of adding the first number. See the first example in `examples.txt`. This needs further examination.) Also,

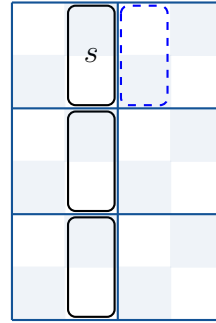
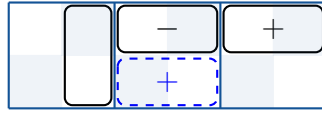
I think after this interchange, we're done here, but I'm not sure. In the code, the function proceeds. So, this also needs further examination. Here is a basic example of the interchange.



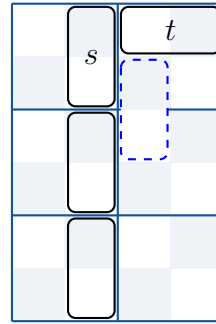
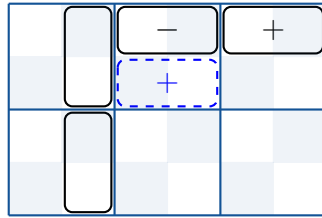
goes to



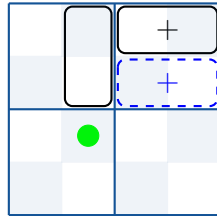
At this point in the function (whether or not we made an interchange) we have a blank, vertical domino as the pair domino. However, that isn't good enough. I think, at this point, there's a sign in the row above. We use `findSignInRowRight()` to find the sign. If it's a $-$ sign, then we're good to go. We can use it to cancel the $+$ sign which we are adding. So, we return this row as the row to add the $+$ sign.



or

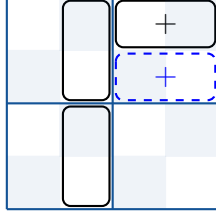


If on the other hand, it's a $+$ sign, then we are going to need another blank domino in the column to the left. We'll need to put the $+$ sign there. If there are no more dominoes in the column, we can't add here. We return the next row to add at.



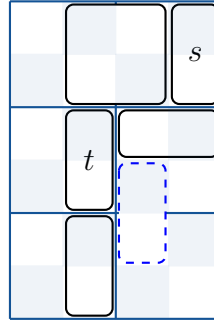
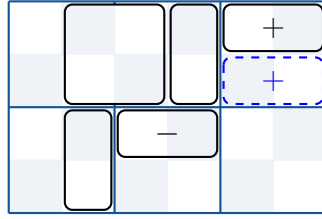
Otherwise, to find out which case we are in, and to prepare for placing the sign, we call `findRowToAddSignX()` at the next row down (and the same column).

Here is a simple positive example.

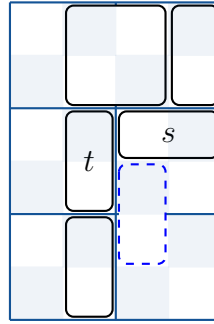
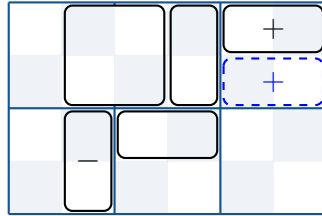


The function `findRowToAddSignX()` doesn't do all the switching which we need for this situation. There may be a domino with a $-$ sign at the bottom of the column, in a horizontal domino. Since the minus sign is not up, we know that we can switch it left, and we do that here. Then, later, we can put a $+$ sign in its place.

Here is an example.



goes to

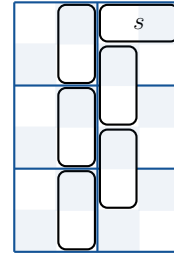
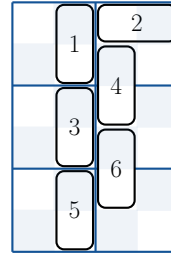
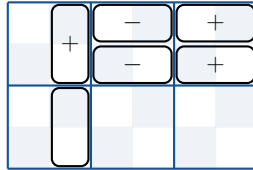
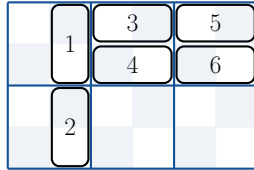
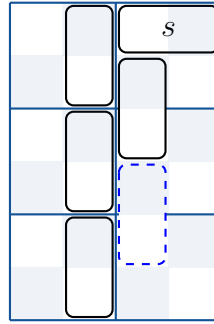
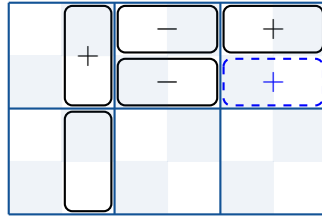


Now we can do the cases for grid position Z . Our top-level cases will be whether the pair domino is horizontal or not.

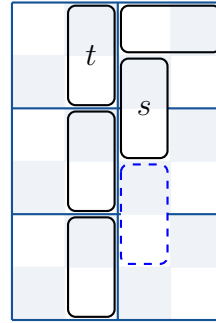
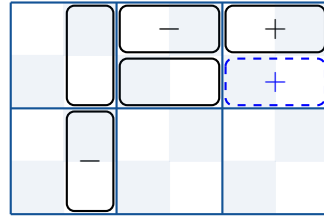
- Here the pair domino is horizontal. The pair domino is then a cycle top. The adjacent domino (to the left) is also horizontal, as part (in

fact an end) of the cycle. Now we break into cases based on the sign in the pair domino.

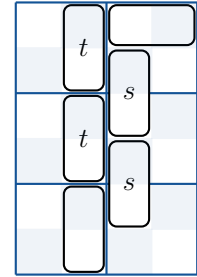
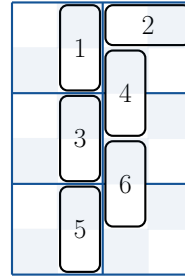
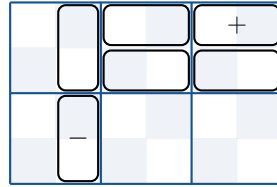
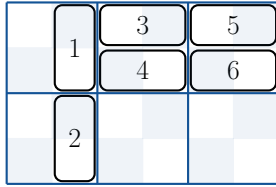
- ▷ Here the pair domino has a $+$ sign. Then the adjacent domino has effective sign $+$ as well, and we can't add at this level. However, the next row down is shorter than this one, so we move there and try to add there.
- ▷ Here the pair domino has a $-$ sign. Then the adjacent domino has effective sign $-$. Its actual sign is either $-$ or blank.
 - * Here the adjacent domino has a $-$ sign. We can just add the new domino here, with its $+$ sign, as a horizontal domino.



- * Here the adjacent domino is blank.



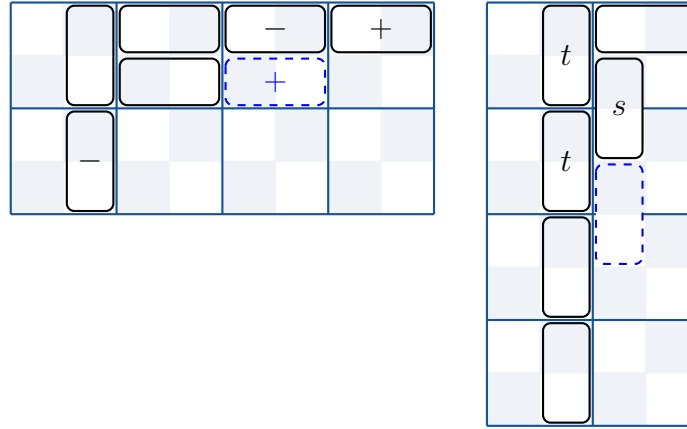
We can still add the domino here, as a horizontal domino. But, this time its sign is on the dual side. We also need to blank out the pair domino on this side, and give the corresponding domino on the dual side the appropriate sign.



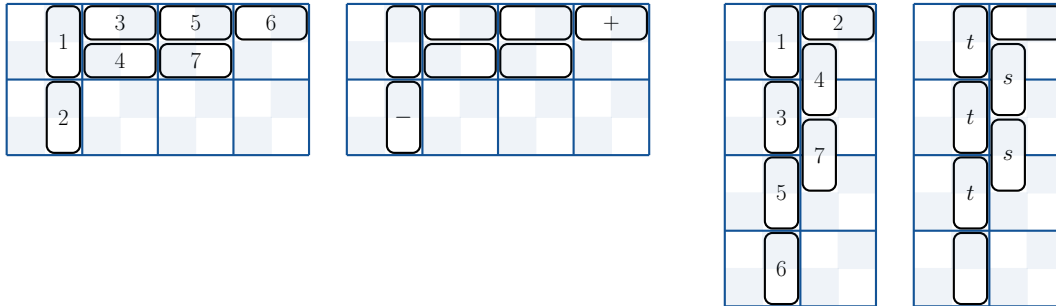
▷ Here the pair domino is blank. However, there must be some sign in the row above, for us to have come down this far. We need to know the effective sign, that is, the leftmost sign. We call `findSignInRowRight()` to get that information.

* Here the effective sign is $+$. Then we can't add the domino to this row. Instead, we go to the next row down.

* Here the effective sign is $-$.



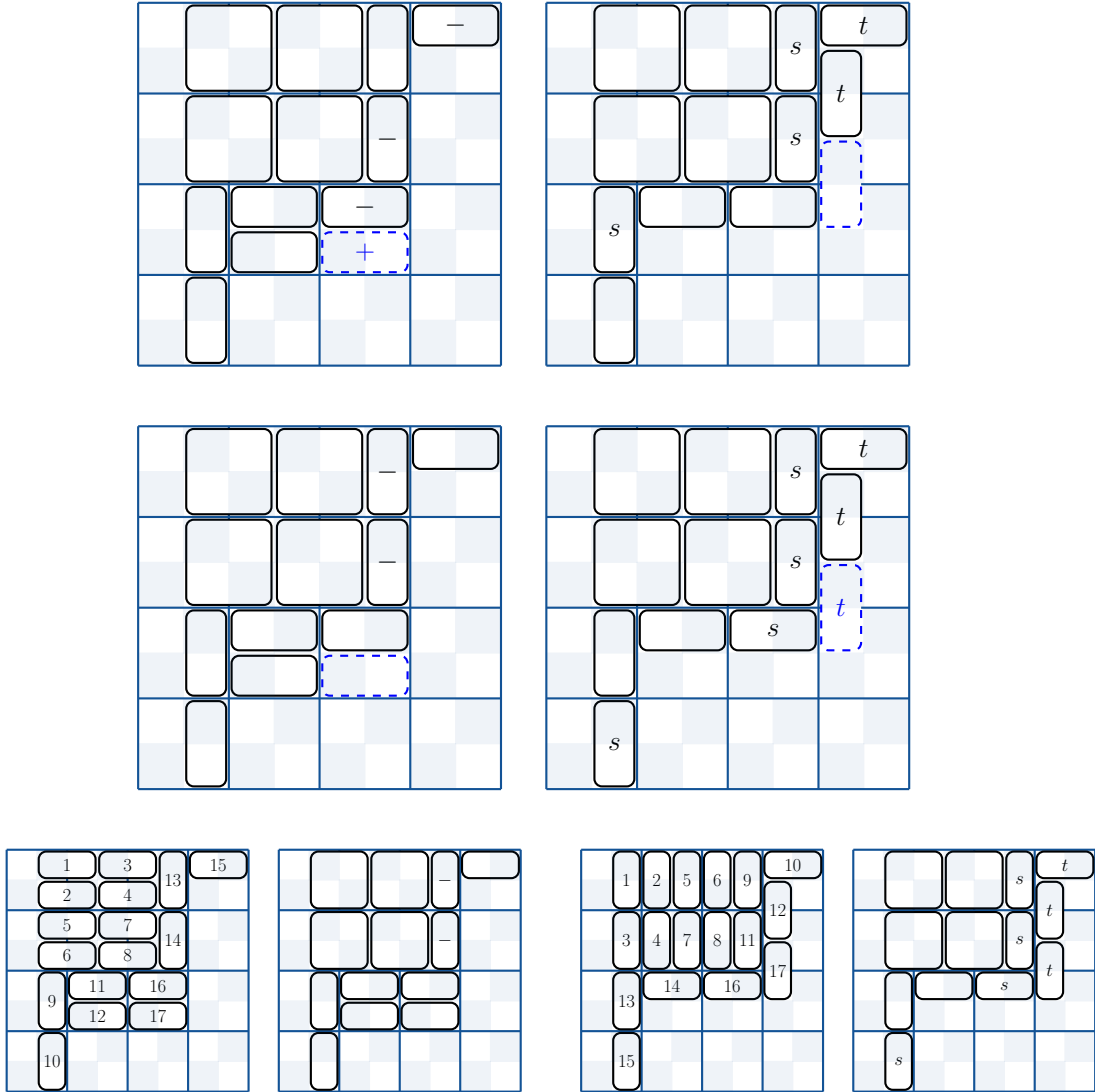
So, we can add the domino at this level. However, it will come in as a blank domino, and we also need to cancel the sign of the domino holding the $-$, and to put appropriate signs into the corresponding dominoes in the other sign tableau.



Since we are putting a sign into the opposite domino of the domino with a $-$ sign, there may be some issues if that domino is horizontal. (Note, it seems like this only happens when that horizontal domino just came down as part of inserting a pair.) We can still place the sign for sure, but we need to handle the issues. The function `prepareForSign()` will do that.

There are two possible issues. On the one hand, the domino may be at the top of a cycle, in which case `prepareForSign()` will call `putSignInCycleTop()`. (Right now, this looks tootly routine, so I won't do an example.) On the other hand, there may be a domino of the same sign in the row. Now, the sign which we are putting in the domino is already

in its column. So, there can't a horizontal domino of that sign in the row, since else it would have moved to the position occupied by the domino which we are putting the sign into. So, the only possibility is that there is a vertical domino with that sign in the row. So, `prepareForSign()` will call `makeSpaceFor()` to move that domino down, if it is there.

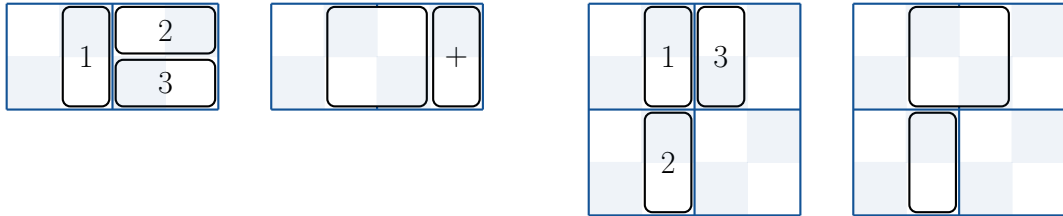
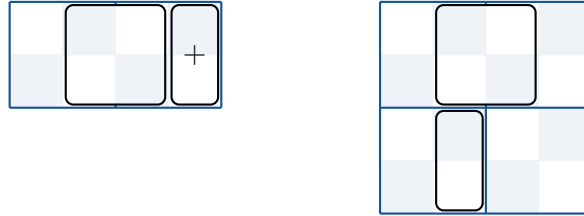
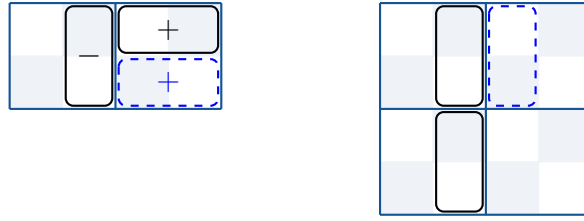


- Here the pair domino is vertical. If the pair domino has a plus sign, we might not be able to add at this level. OTOH, we might be able

to move such a domino down. The function `findRowToAddSignZ()` takes care of the situation. Based on its output, we may have to add at a lower level.

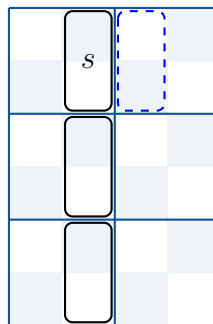
Henceforth, assume we can add at this level. The two cases depend on the length of the next row.

- ▷ Here the next row has a smaller length than this row. So, we are contracting a type I cycle. The two cases depend on the sign in the pair domino.
 - * Here the pair domino has a $-$ sign. We just put the domino and then box it up.

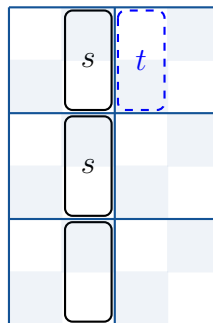
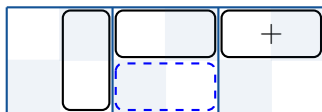


- * Here the pair domino is blank. Then, there is a $-$ sign in the row above, as the leftmost signed domino. (TODO, check why this is true.) We're going to cancel its sign, which means placing a sign in the corresponding domino on the dual side. So,

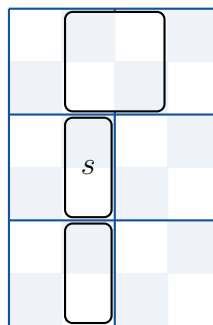
we need `prepareForSign()`. After that, we place dominoes and box them up.

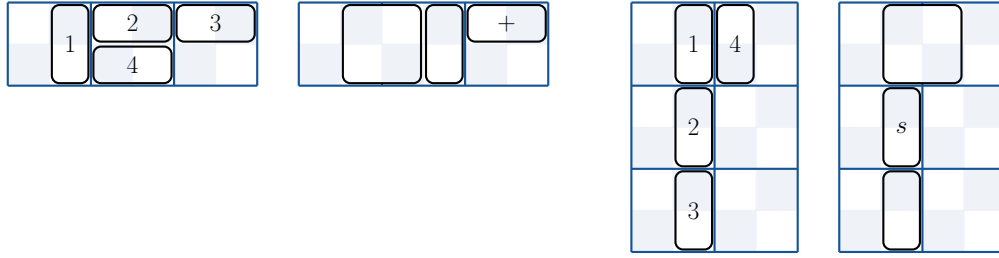


goes to



goes to

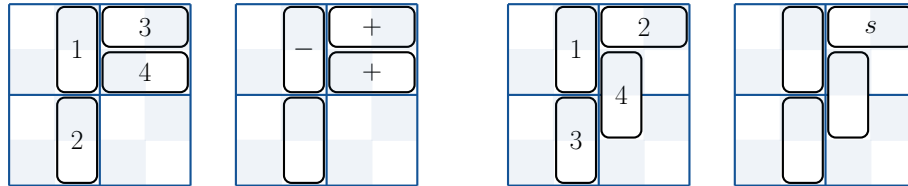
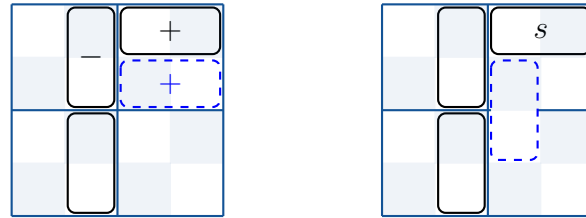




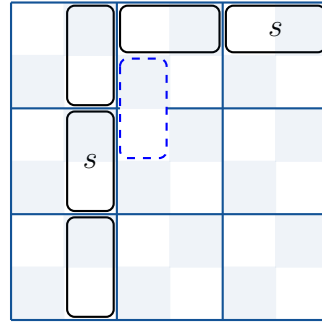
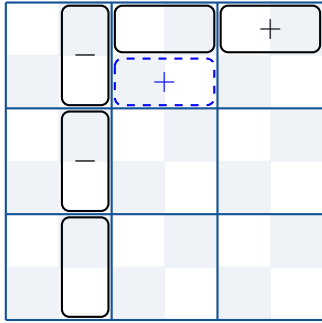
▷ Here the next row has the same length as this row. So, we are making a new type II cycle. The two cases depend on the sign in the pair domino.

* Here the pair domino has a $-$ sign. So, basically, we can put the domino here, with a $+$ sign, since that matches the pair domino. However, the configuration may not then be an allowed one for paired cycles, so we may have to adjust for that. This depends on what is in the domino just above, and in the domino just below the pair domino. Here are the cases.

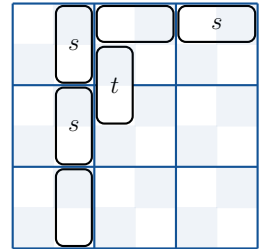
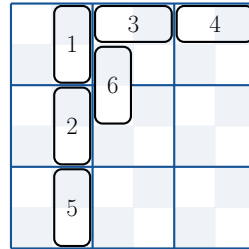
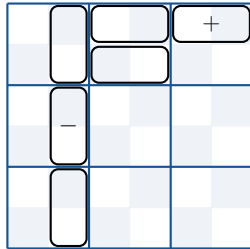
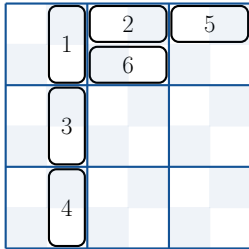
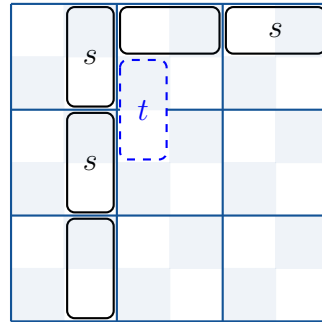
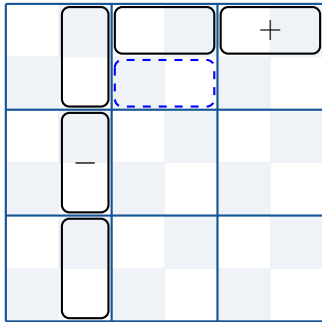
⌘ If the domino just above has a $+$ sign, we can just put the new domino here, with a $+$ sign.



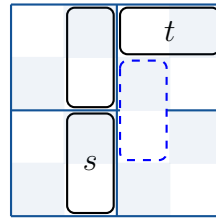
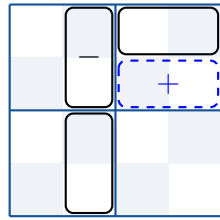
⌘ If the domino just above is blank, we need to check further. If the domino just below has a $-$ sign, we can just blank the new domino and the pair domino, and put the appropriate signs in the other side.



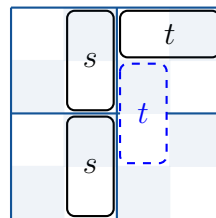
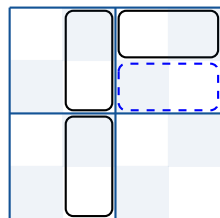
goes to



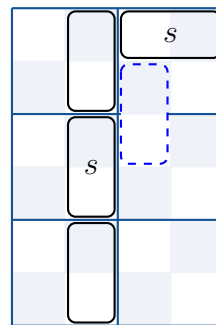
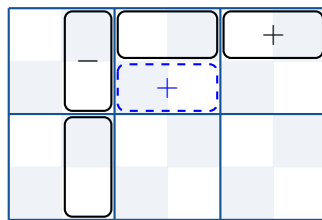
Similarly if the domino just below is blank, and the corresponding domino on the other side has a the opposite sign to the domino on the side in the dual tableau. (However, it looks like this case doesn't occur. TODO, check this. Hopefully, find the theoretical reason why this doesn't occur, and then take it out of the code.)



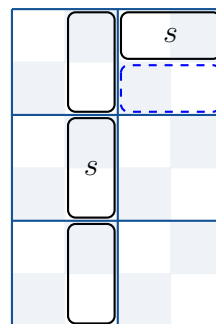
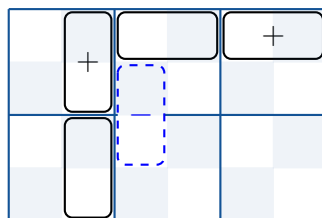
goes to

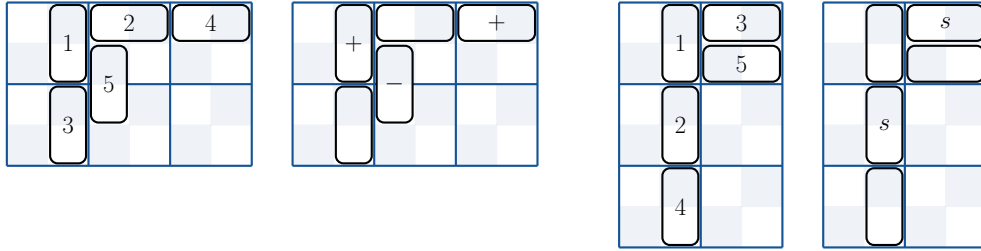


Otherwise, we need to shift the new dominoes and flip their signs on this side.

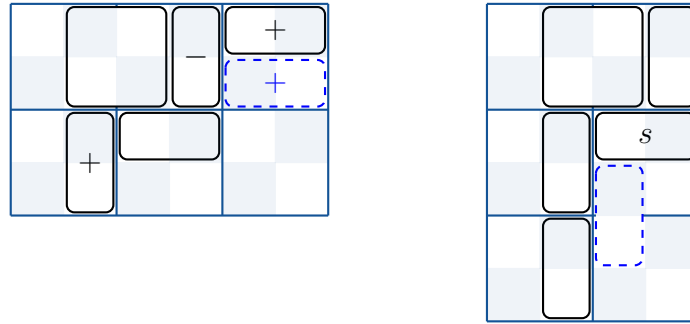


goes to

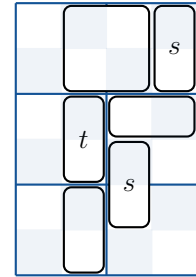
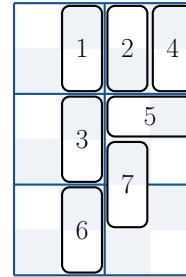
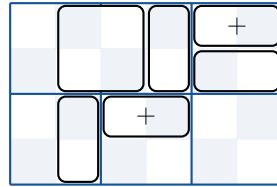
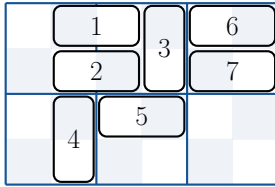
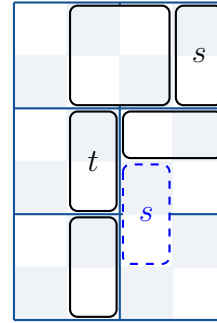
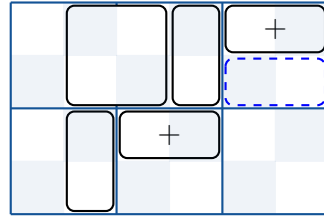




Having done this, we notice that a we have created an opportunity for dominoes to move, since the $-$ sign in the pair domino has now been neutralized by the newly-added domino. That is, if the rest of the column in the left tableau is blank, and if the bottom of the column is horizontal, there may be a $+$ sign to the left of that bottom domino which can now move into the bottom of the column. Concretely speaking, `switchAfterAddZ()` is called on the dual domino to the blank bottom corner domino. Note, moving one domino may make room for another domino to move up. The function `switchAfterAddZ()` calls itself (sometimes) after making the one switch.

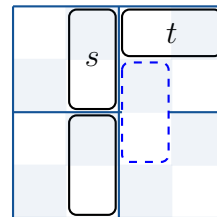
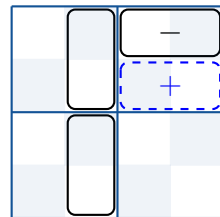


goes to

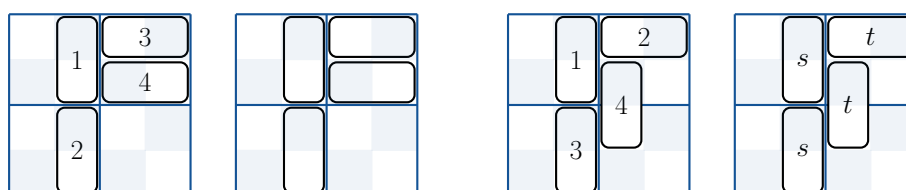
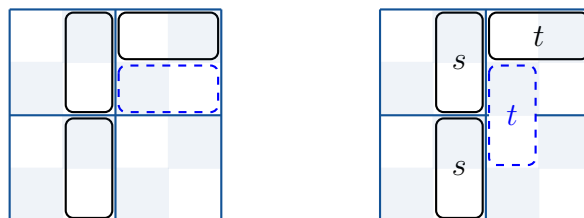


* Here the pair domino is blank. There are a number of cases. In them we have a top domino, which is above the domino which we are trying to add, and a side domino, which is directly below the pair domino. We also have these on the dual side, that is, the dual top domino corresponds to the side domino, and the dual side domino corresponds to the top domino.

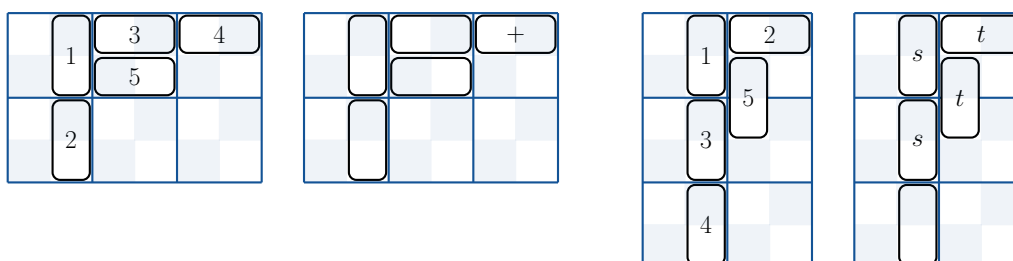
⌘ Here the top domino contains $-$. Then we are basically good to go. We'll blank the top signs and fill them on the dual side. First, we'll call `prepareForSign()` on the dual side, so as to be able to place a sign in the dual side domino.



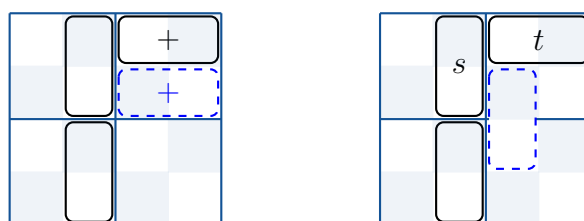
goes to



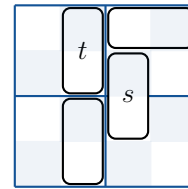
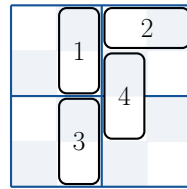
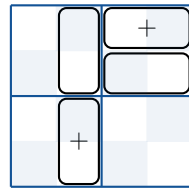
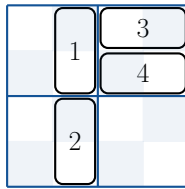
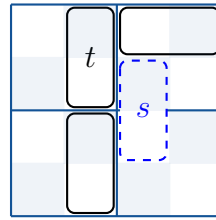
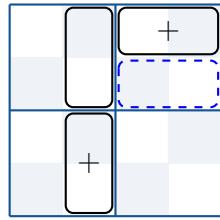
Or, starting with one more + domino in the top row, we get



✕ Here the top domino contains +. (Note, in the code, this is the third case.)

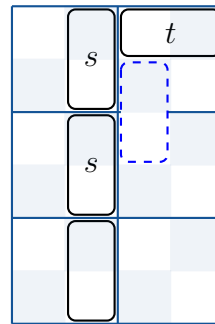
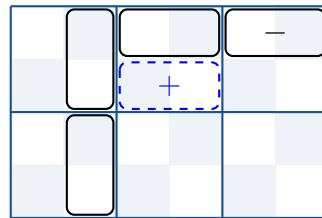


goes to

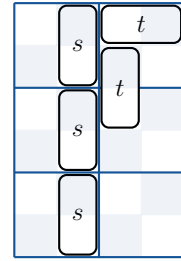
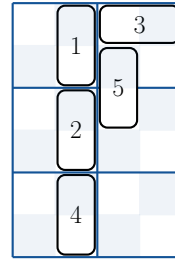
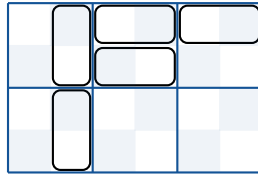
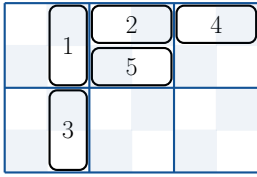
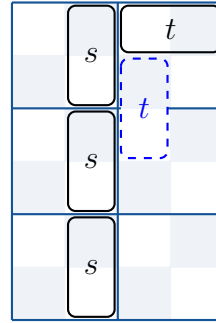
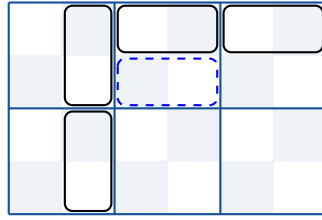


- ⌘ Here the top domino is blank. (Note, in the code, this is the second case.) We need to find the sign in the top row. (TODO, there must be a sign in the top row. Why?) There are two cases.

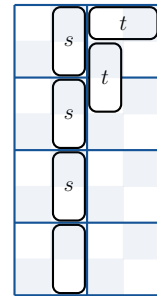
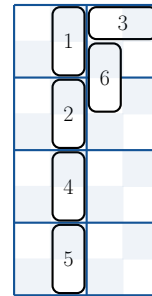
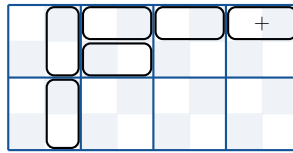
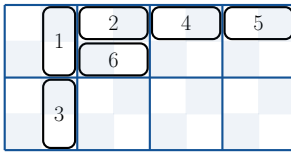
Here the sign in the top row is $-$. This is basically the same as the situation where the top domino is $-$.



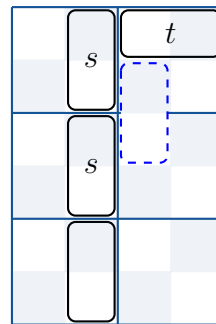
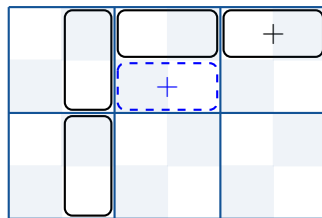
goes to



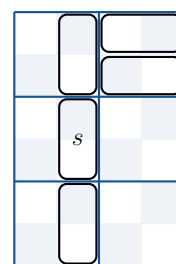
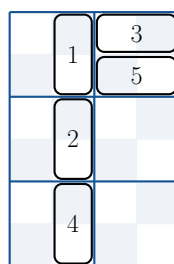
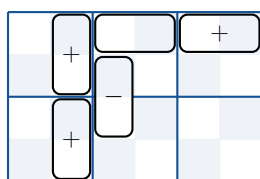
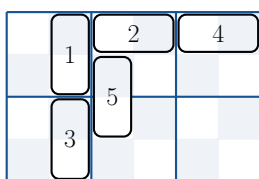
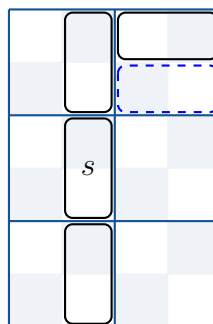
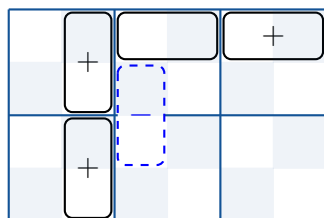
Or, with an extra + domino in the top row,



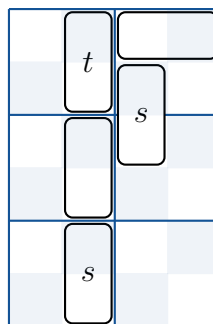
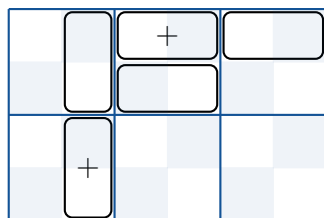
Here the sign in the top row is +.



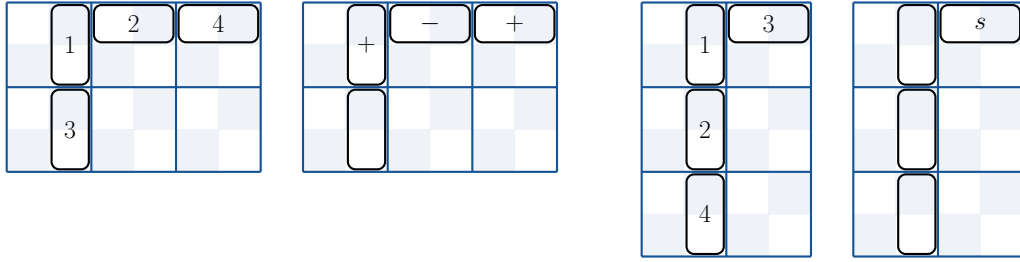
goes to



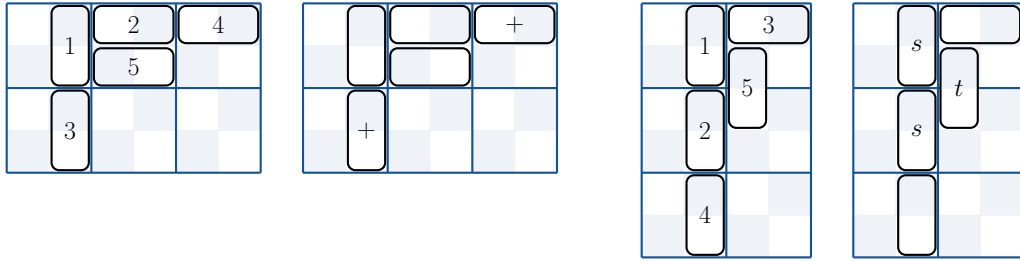
So, how do we think about this outcome? We can think that this is related to the outcome when $+$ is in the top domino. That is, if we interchange the two top dominoes, then, seeing two $+$ dominoes together, the cycle would respond by moving up. It would look like this.



It also might help to look at the input which leads to the conjugate result, that is, with the cycle up. It's this:



goes to



To conclude the discussion of the case where the pair domino is vertical and blank, we note that, as in the case when the pair domino is vertical and has a minus sign, there may now be an opportunity for a domino to move. This time, we have that an s (or t) in the pair domino on the dual side has been neutralized. So, we may call `switchAfterAddZ()` on this side.