OPERATIONS WITH DECIMALS

Since decimals are a type of fraction, it shouldn't surprise you that models for operations with decimals are very similar to those for fractions. Remember that models are limited in scope—the idea is to build the concept to scaffold students to models with more complicated numbers.

Adding Decimals

If we're multiplying a fraction by a whole number, we can use our traditional strategy: make the given number of copies of the fraction, then combine them to see what the total is.

Consider **0.3 + 0.25.**
Like every other model for addition, represent each of the numbers, then combine the shaded regions. Just like with fractions, it's easier with a common denominator so that our pieces are the same size. So, we will replace 0.3 with the equivalent 0.30 by subdividing each region into ten more.

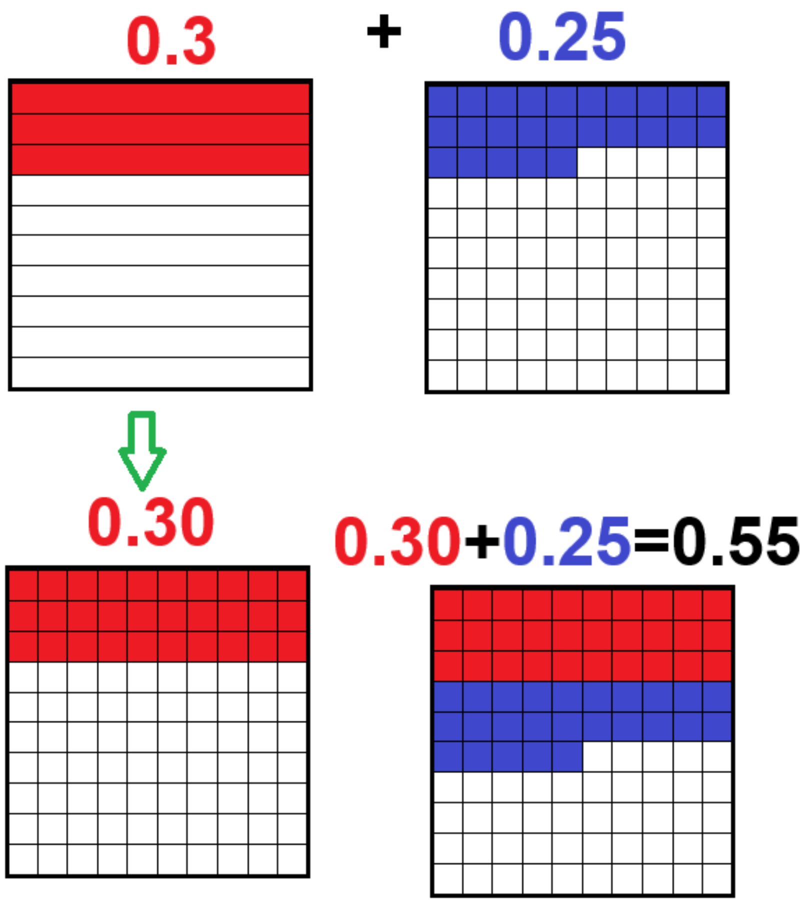


Figure 1: 0.3+0.25

What if the decimals are bigger than one? The model is the same, but you have to make sure you understand where the “unit” or 1 is.

Note in the example below that the unit is defined by a darker box around the 100 grid. Consider **1.32+1.49**. We shade one whole and 0.32 more orange. In the next box we shade one whole and 0.49 more green. When we combine the shading, we have 2 wholes and .81 left.

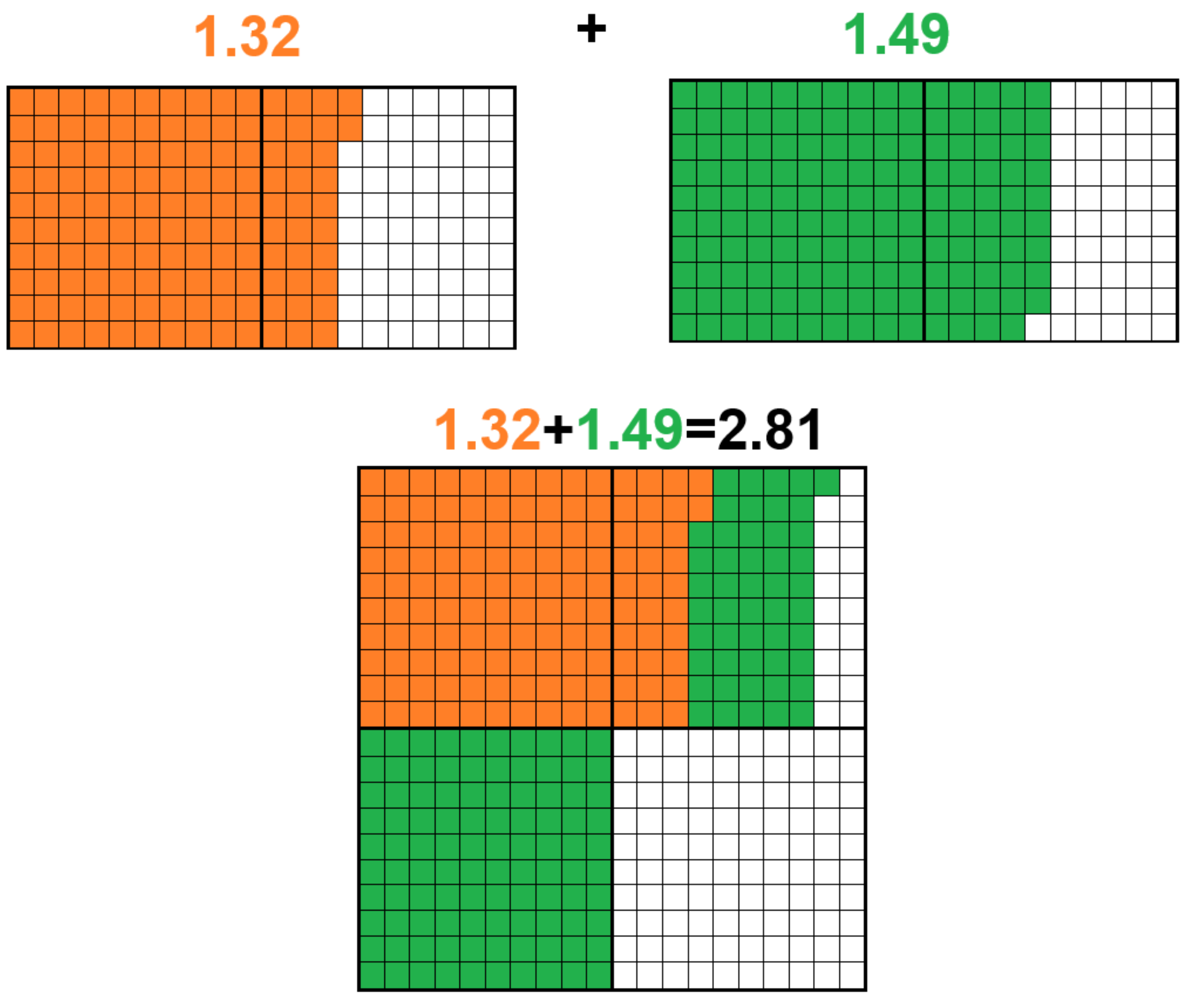


Figure 2: 1.32+1.49

How to we connect this to the “traditional” algorithm? The big idea when adding fractions is to line up the decimal points, then add. ***Why do we line up the decimal points?*** This is making sure we’re adding hundredths to hundredths, tenths to tenths, and so on. We’re making sure we’re combining numbers that are the same size. You could also show this with base ten blocks, as long as you redefine your one. (See Models for Decimals.) This lets you model regrouping (borrowing or carrying) by trading pieces up or down.

Subtracting Decimals

Like before, we represent the first number, then take away the second. Depending on the problem we may need to rename one fraction with an equivalent fraction so the pieces are the same size. Like with addition, we “line up” the decimal point before subtracting so we’re dealing with similar size pieces.

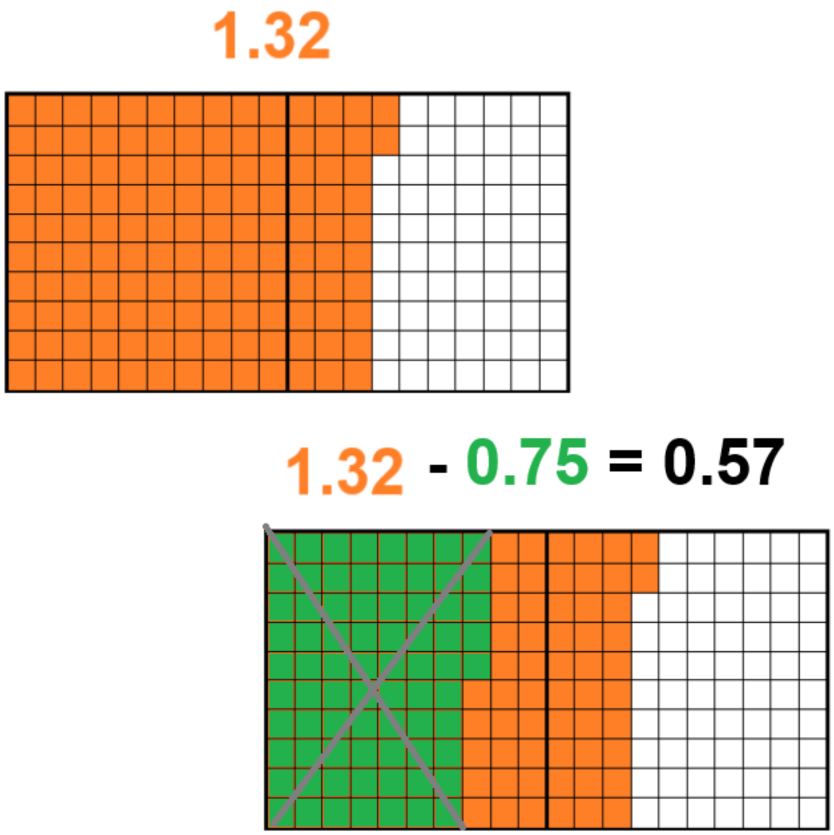


Figure 3: 1.32-0.75

You can also connect with the traditional algorithm using base ten blocks to illustrate borrowing and carrying. Note that lining up the decimals is still important in subtraction. Note in the example below that the one is represented by a flat, tenths by longs, and hundredths by units.

See if you can follow how the subtraction works from what we’ve previously discussed.

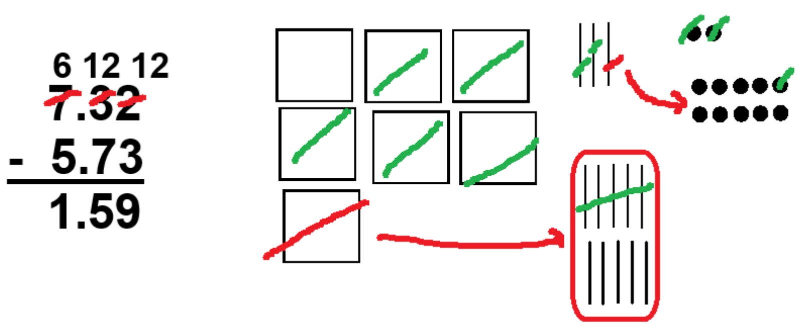


Figure 4: Subtracting decimals with base 10 blocks

Multiplying Decimals

Models for multiplying decimals are based on the model for multiplying fractions: Shade one decimal vertically and another horizontally, and see where they overlap.

Consider **0.7 x 0.4**:

We shade 0.7 in blue horizontally and 0.4 in red vertically. When they overlap, we see that 0.28 is double shaded in purple.

Why does tenth x tenth give hundredths? Think about it in terms of fractions:

$$\frac{7}{10} \times \frac{4}{10} = \frac{28}{100}$$

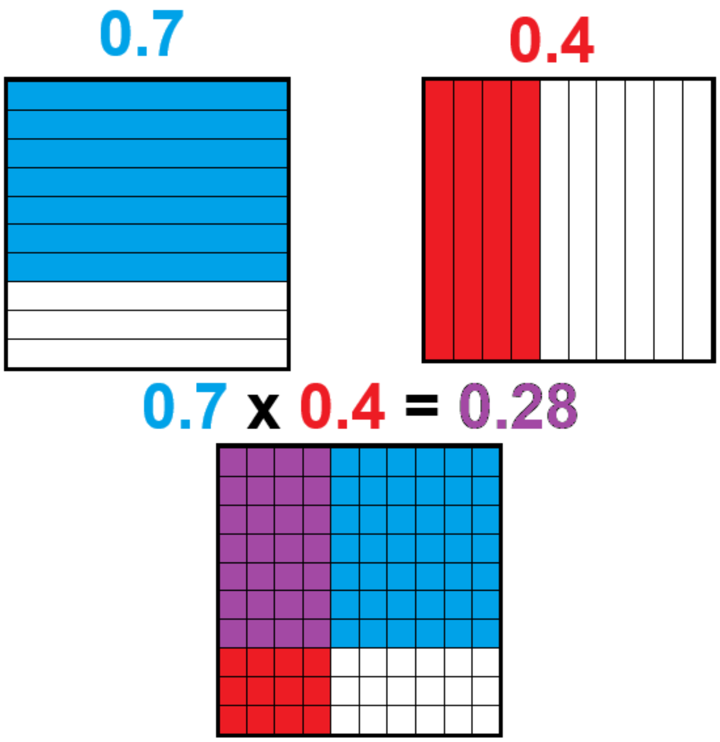


Figure 5: Decimal multiplication

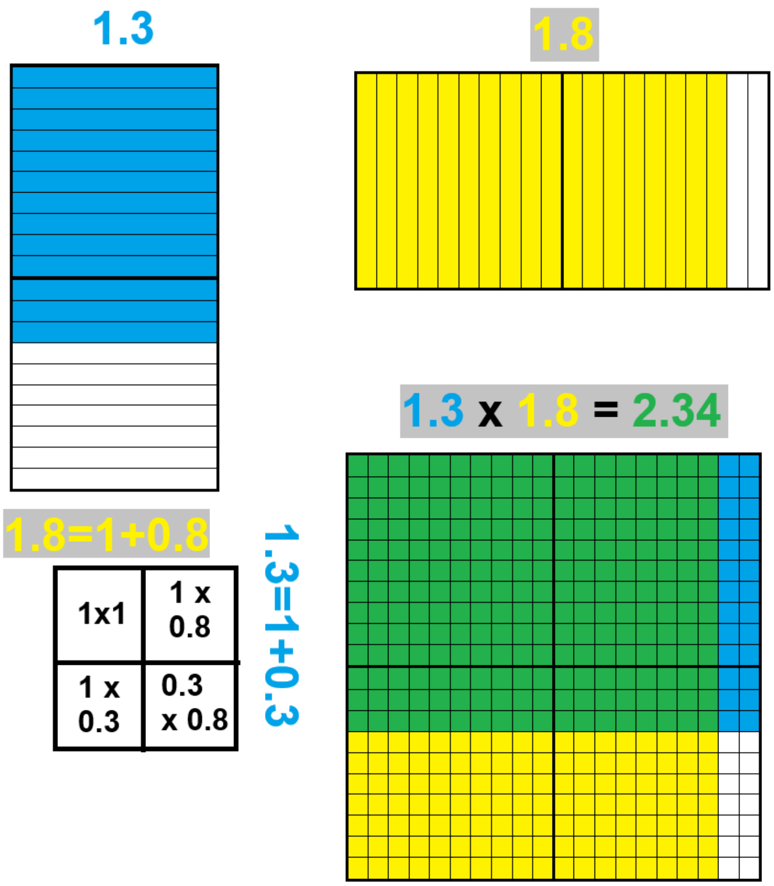
Think about the traditional algorithm. You multiply the numbers, then count the decimal places to see how many are in the answer. That works because you're multiplying the denominators together to get the new number of boxes, just like we did with fractions.

Note that this works with smaller place values, too. It's just a lot harder to draw a thousandths grid, so we'll keep our examples on the small side.

When the fractions are bigger than one, it gets a little more complicated. We need to be clear where 1 is represented. We also need to make sure every part of the first fraction gets multiplied by every part of the second fraction. Think back to the area model for multiplication, as well as what the model looked like in fractions.

Consider **1.3x1.8**:

When we shade 1.3, we need two boxes to represent our number: 1 whole and 3 tenths. (This is equivalent to 13 tenths.) When we shade 1.8, we also need two boxes to represent our number: 1 whole and 8 tenths (also known as 18 tenths). That means when we do our multiplication, we need 2x2 or 4 grids to represent our answer. We have to multiply every part of 1.3 by every part of 1.8. So we get 1x1=1 in the top left, 1x0.8=0.8 in the top right, 1x0.3=0.3 in the bottom left, and 0.3x0.8=0.24 in the bottom right. Adding all of that together, we see that we have 2.34 shaded green where the numbers overlapped.



$$\begin{array}{r} {}^{2} \\ 1.3 \\ \underline{\times\ 1.8} \\ 104 \\ \underline{13} \\ 2.34 \end{array}$$

Ignore the decimals and multiply 13x18. 2 decimal places in the problem so 2 in the solution (tenths x tenths = hundredths)

Figure 6: Standard algorithm

Figure 7: Decimal multiplication with numbers bigger than 1.

Dividing Decimals

The idea behind dividing decimals is the same as that for fractions. How many of the divisor fit into the dividend?

Consider **1.5 ÷ 0.2**. How many copies of 0.2 fit inside 1.5? We see 7 complete sets fit, with 1/2 boxes left over. So the answer is 7.5.

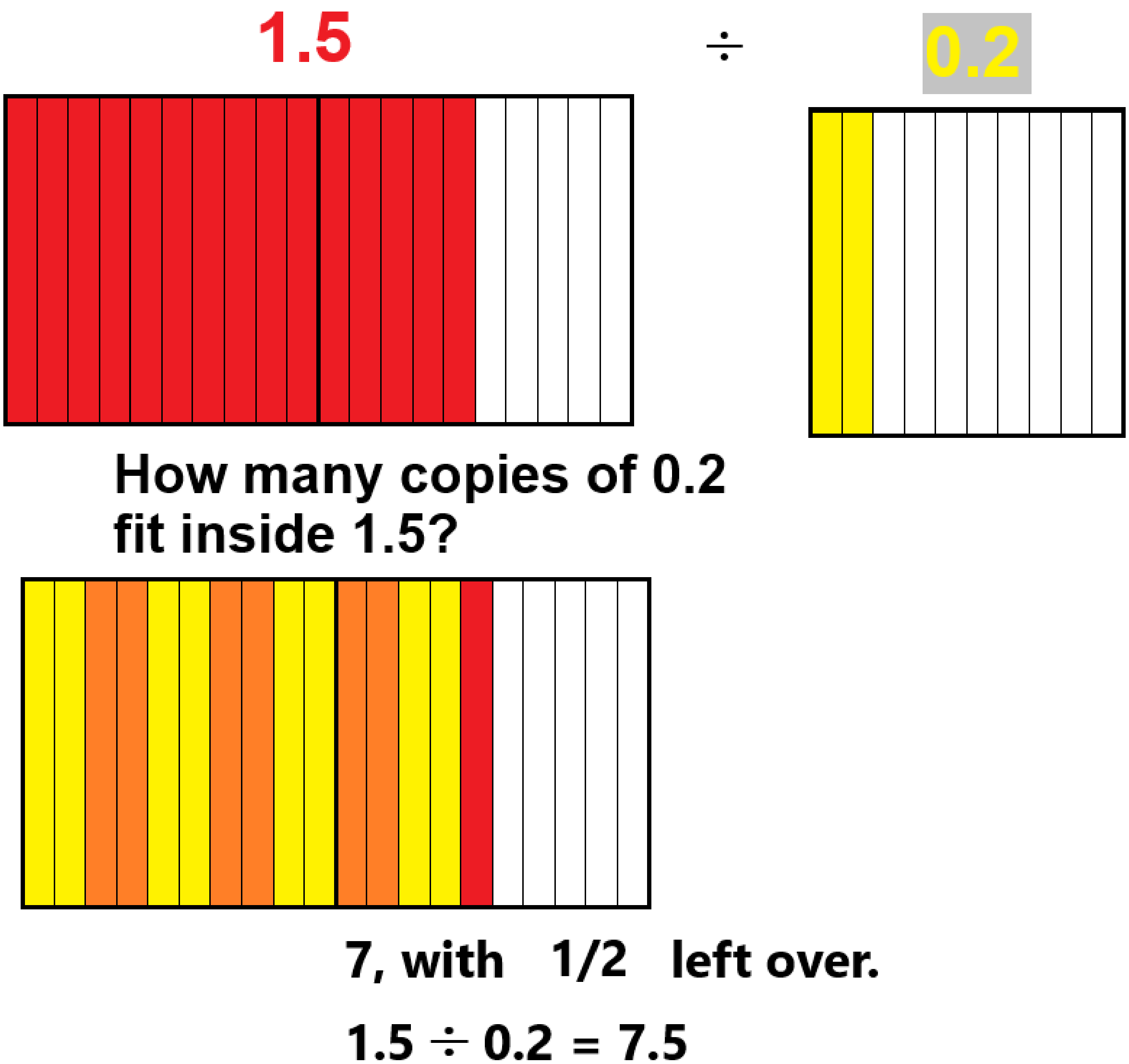


Figure 8: 1.5 divided by 0.2

The traditional algorithm says we can move our decimal point so the divisor is a whole number, provided we move the decimal point the same number of places on the dividend. Why doesn't that change our solution?

$$0.2\overline{)1.5} \Rightarrow 2\overline{)15}$$

$$\frac{1.5}{0.2} = \frac{1.5}{0.2} \times \frac{10}{10} = \frac{15}{2}$$

Figure 9: Moving the decimal point

Basically, it's because we're finding equivalent fractions. Remember that a fraction is a division problem, so we can write our problem as 1.5/0.2. I can find an equivalent fraction by multiplying the numerator and denominator by the same number, in this case 10. (Note that 10/10 = 1.) So our problem is equivalent to 15/2.

Note that when dividing decimals, you should always divide until your answer terminates (stops) or starts repeating. Remainders don't usually mix with decimals.

More on dealing with repeating decimals in the Conversions section.