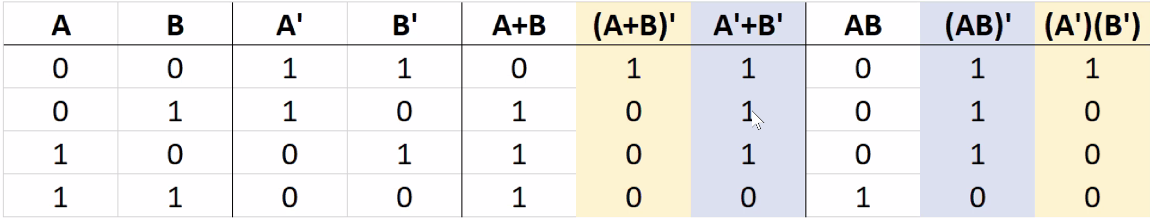
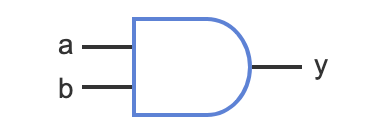
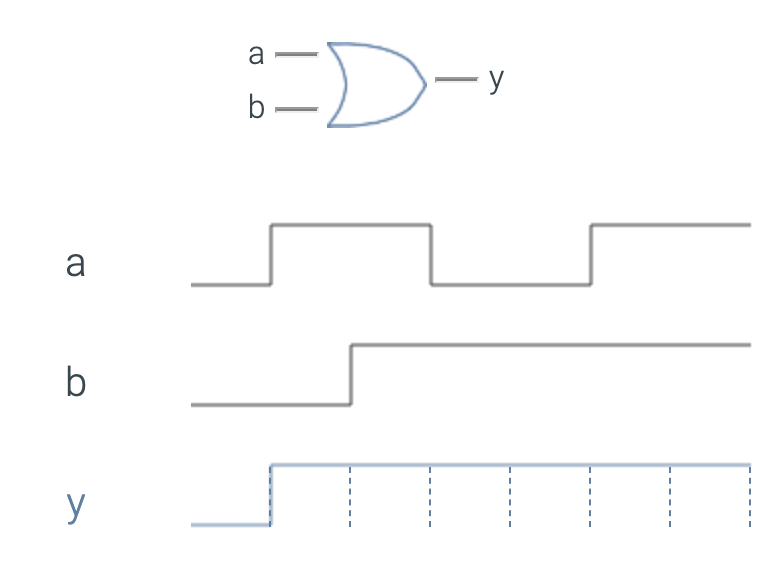
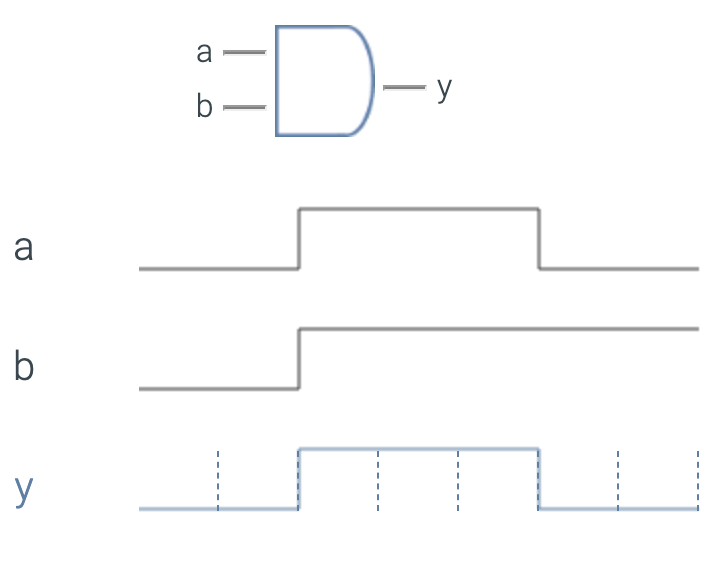
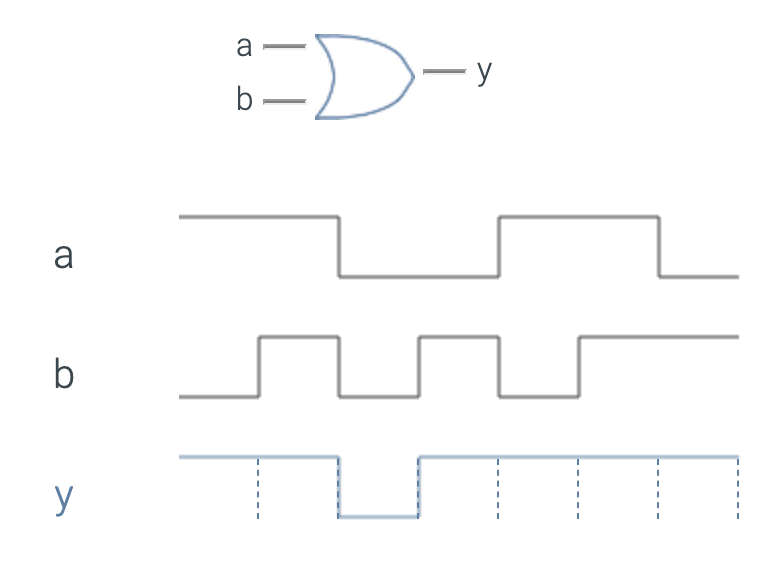
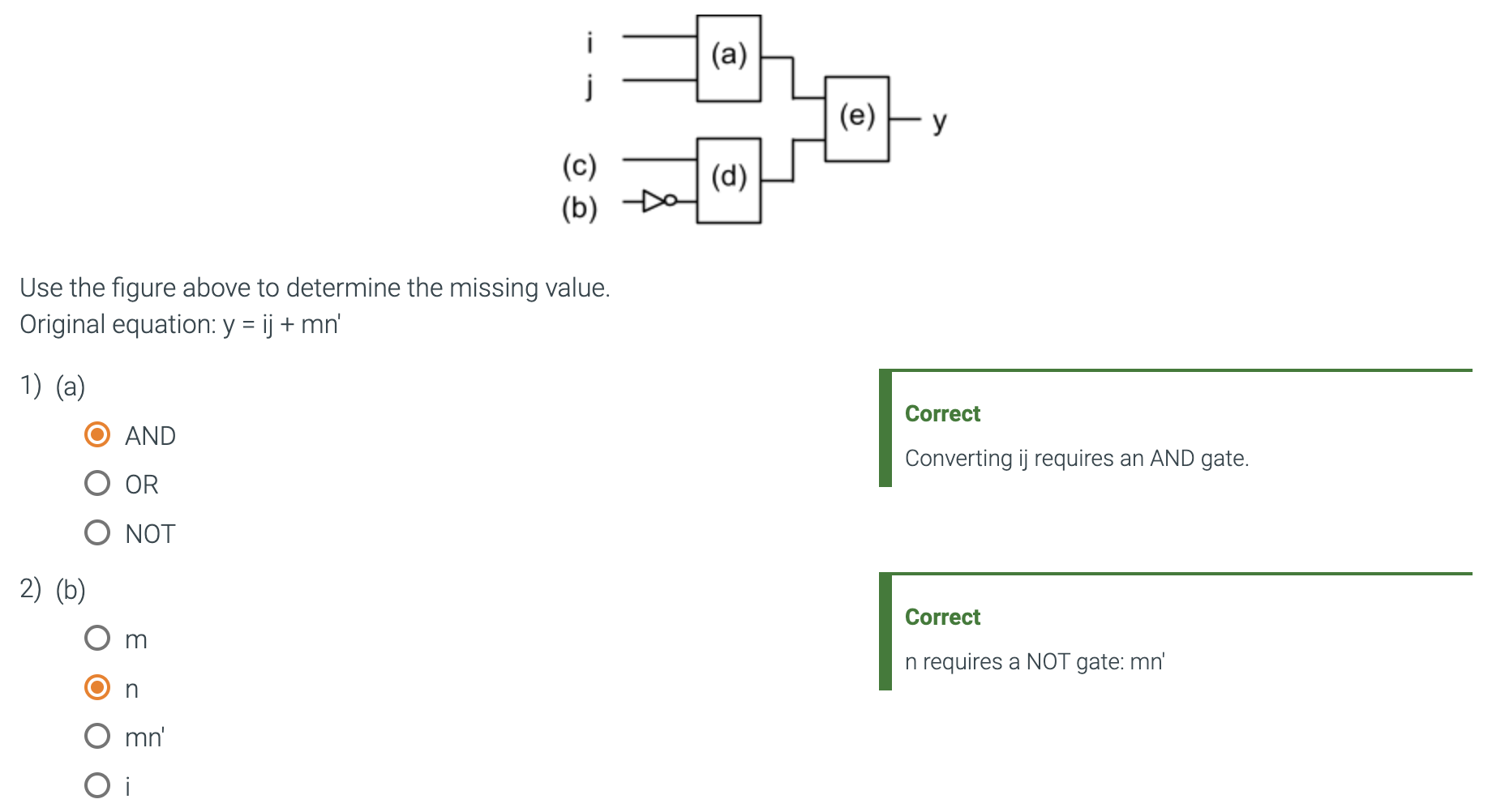
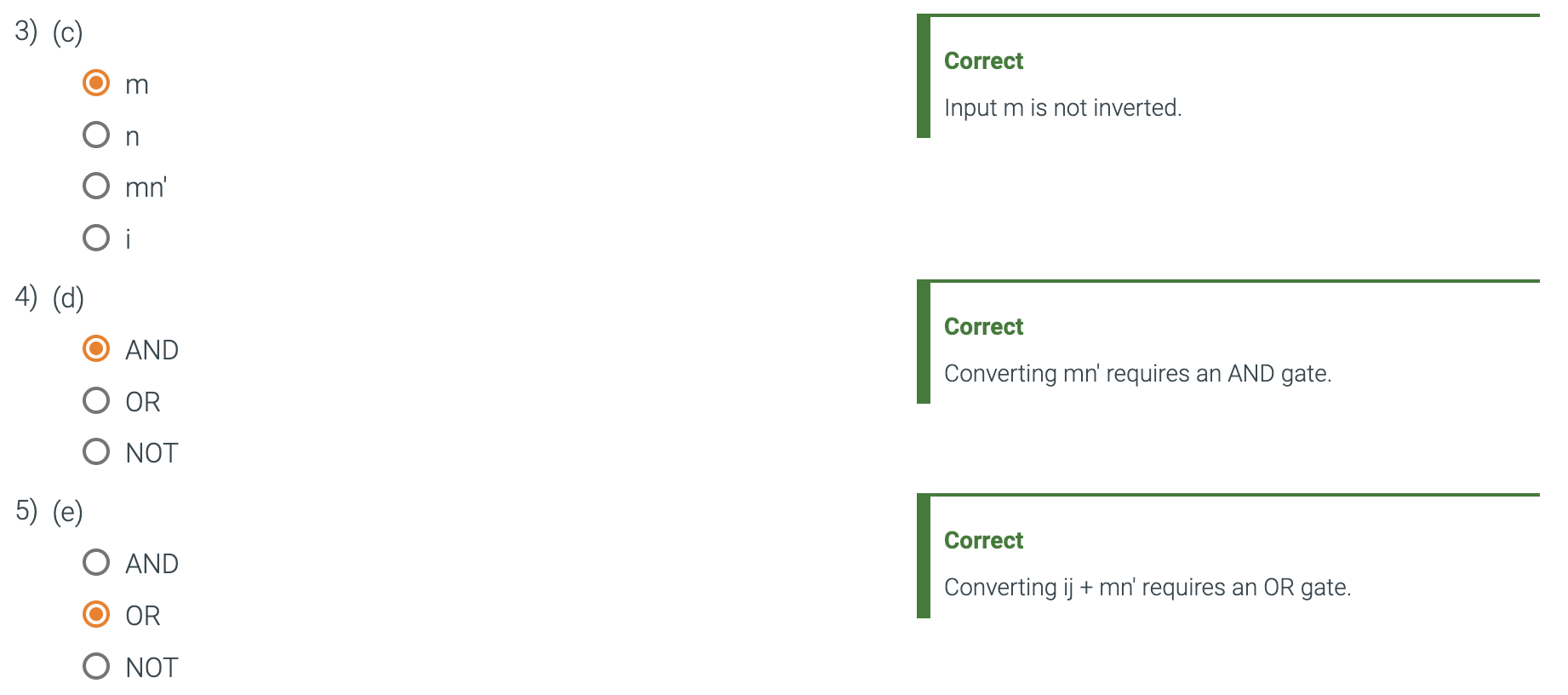
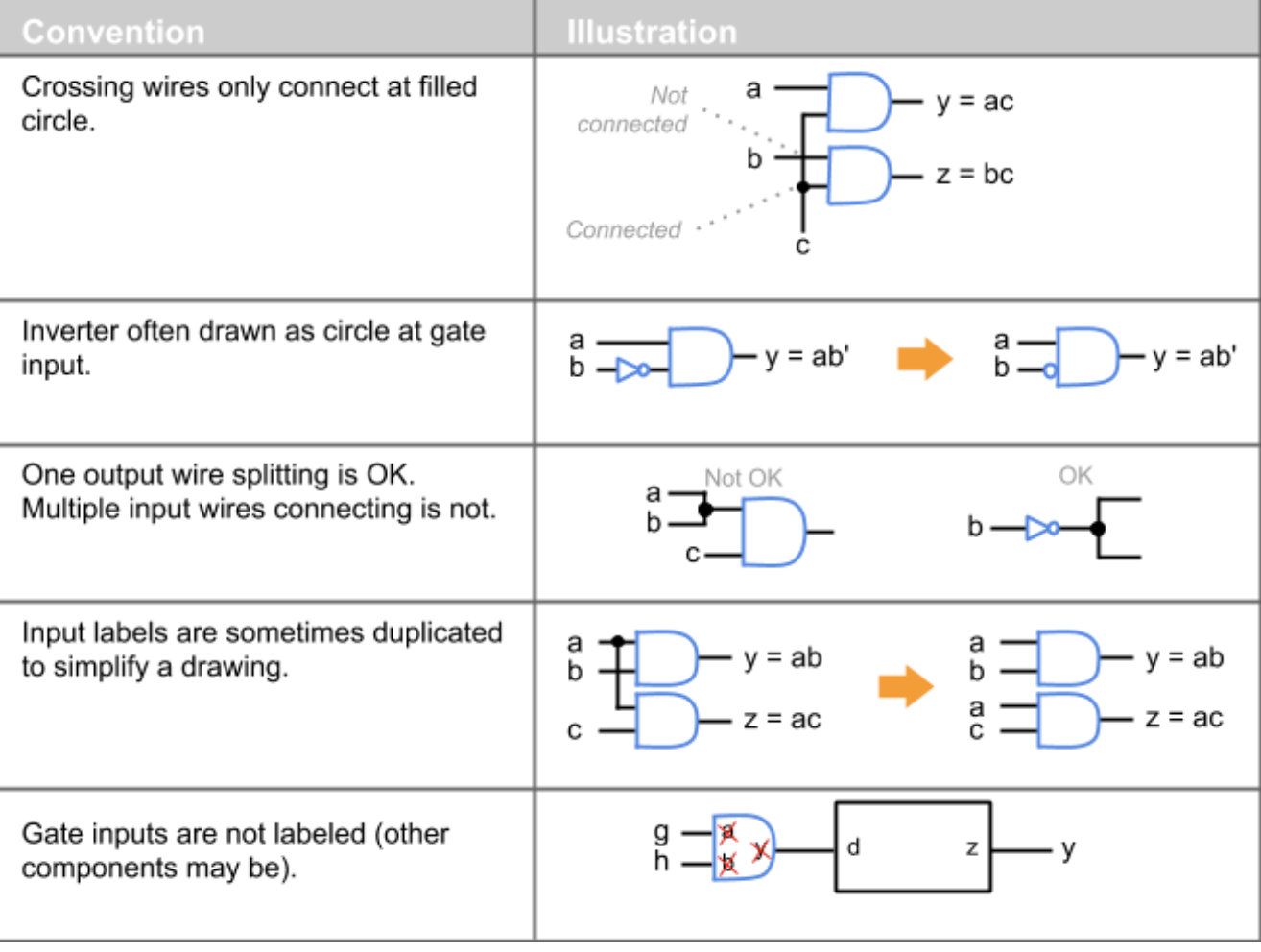
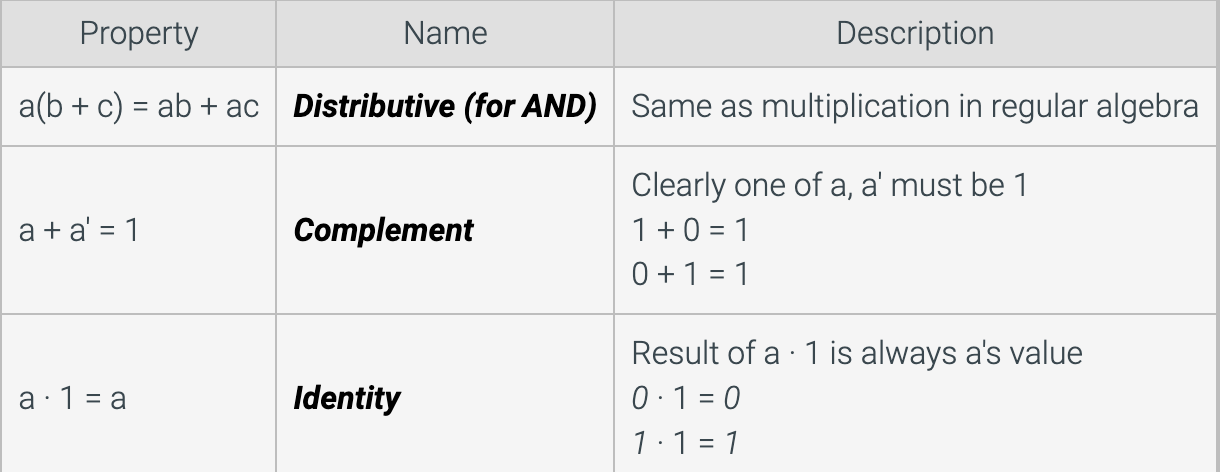
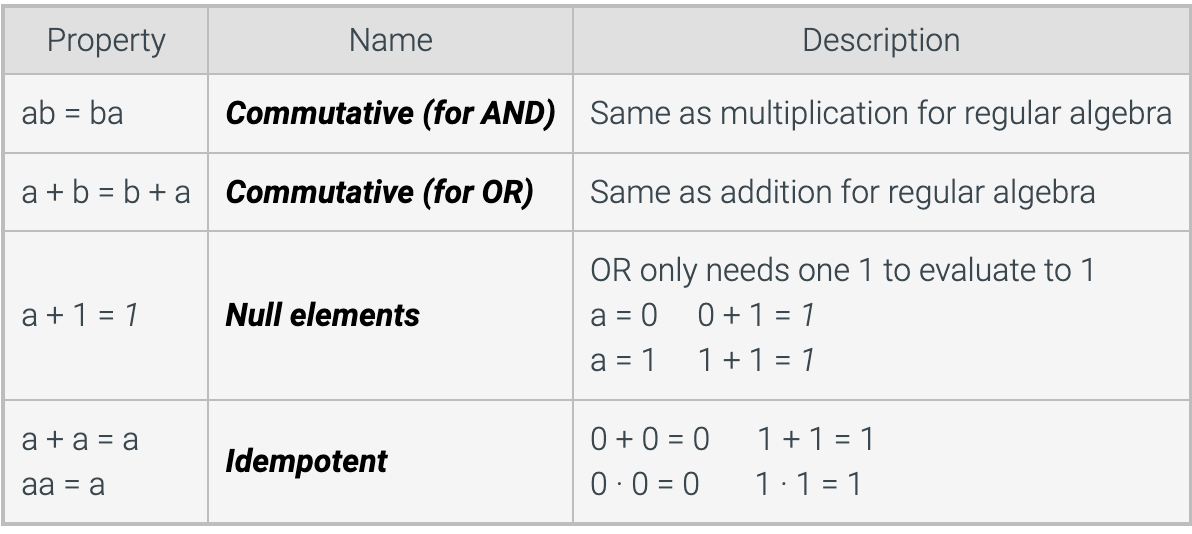
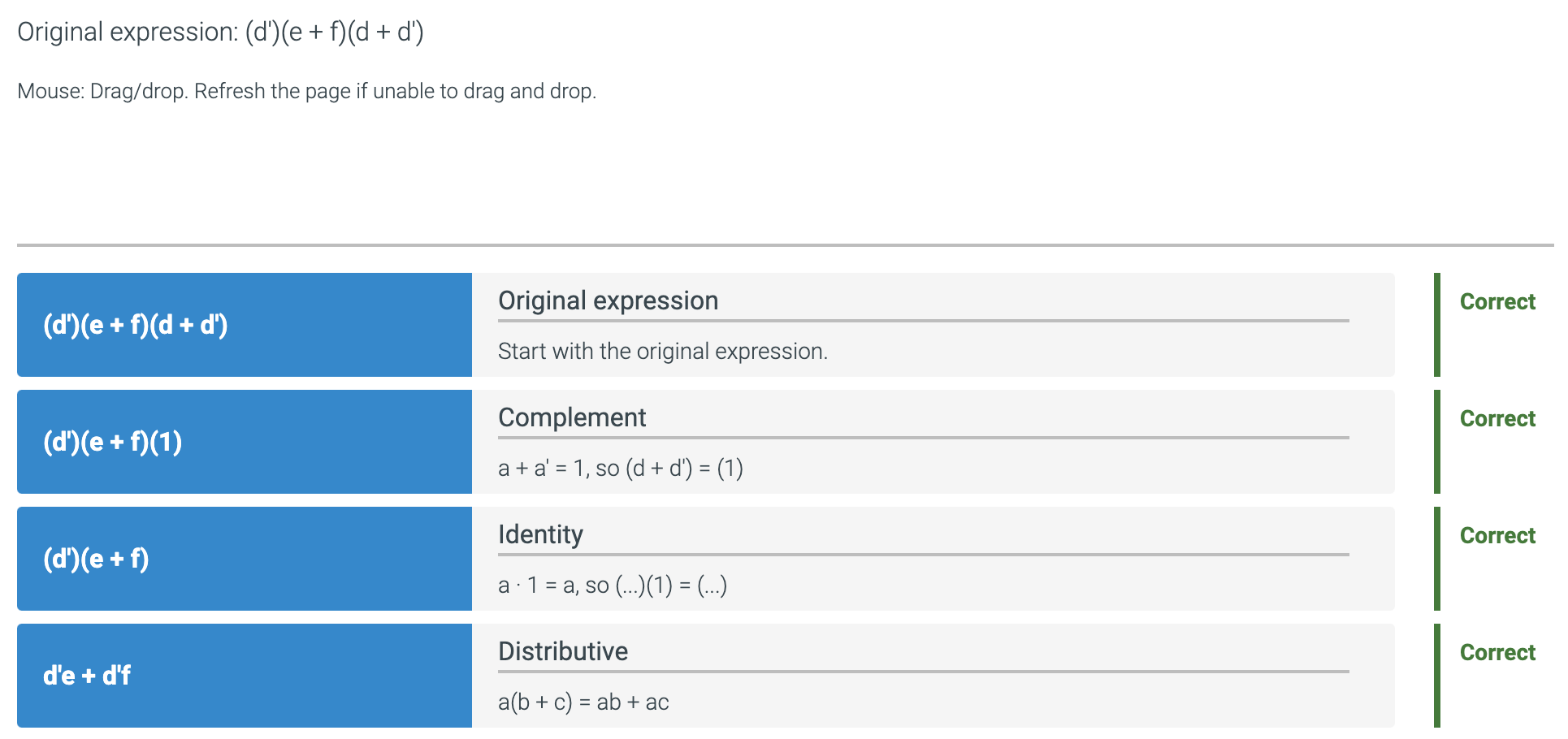
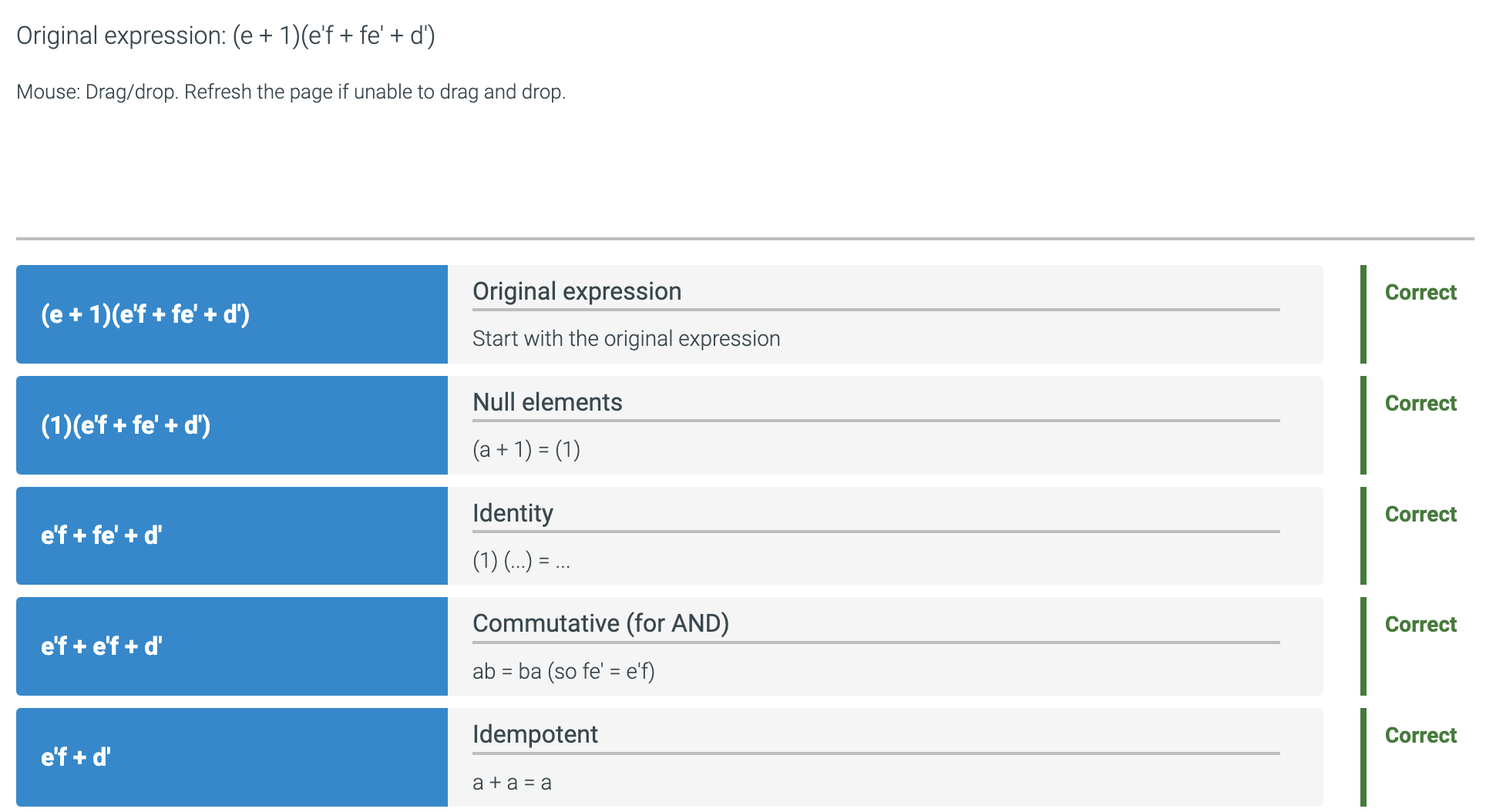
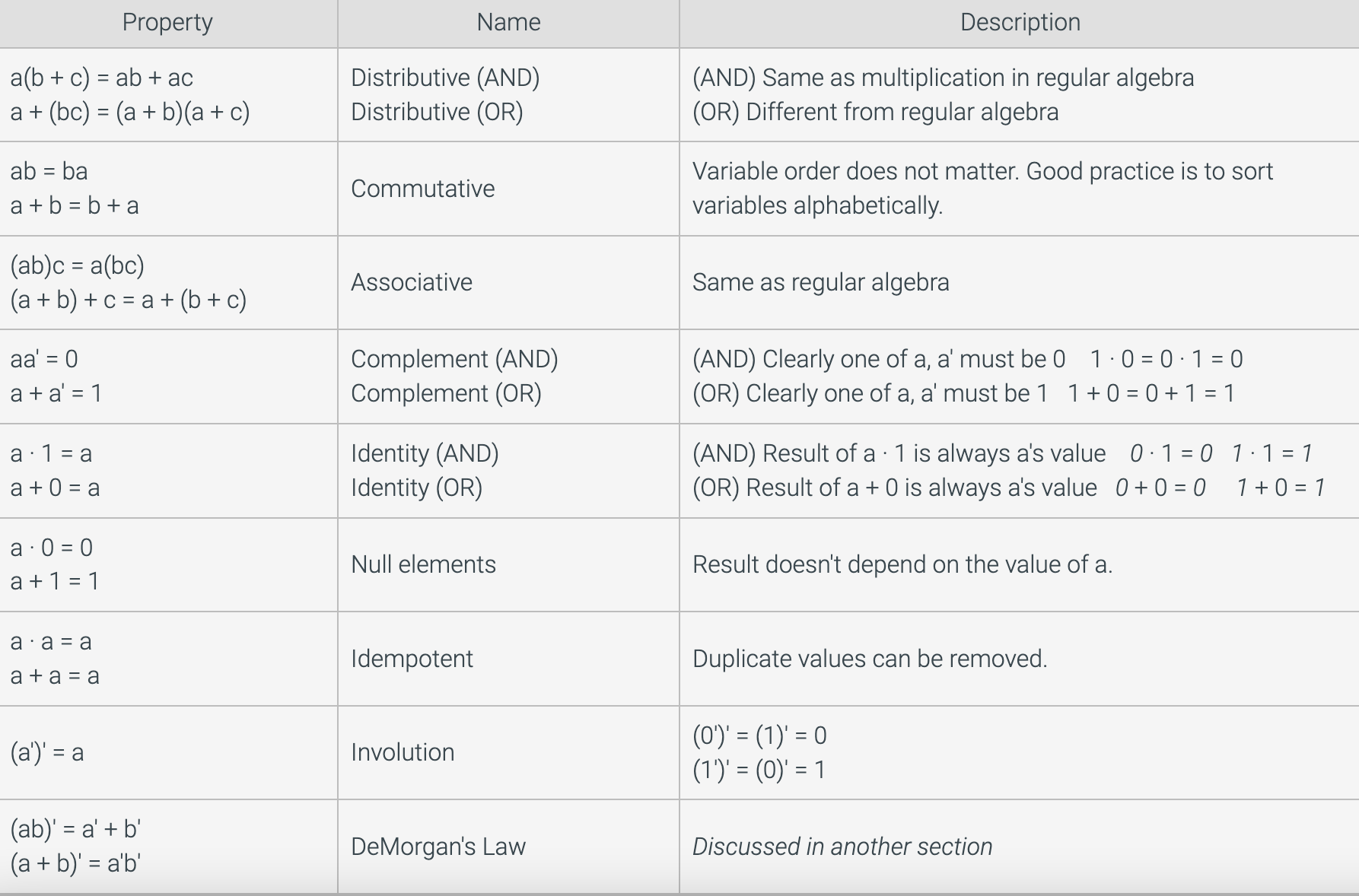
**Chapter 11.5-11.10 Notes**

* Lecture Notes
  + Circuit timing caveat
    - Transistors are analog devices with non-zero switching times
    - Unbalances circuit paths can lead to issues
    - Glitch
      * Undesired signal lasting only a short time
    - Race Condition
      * Output depends on small differences in signal timing
    - CAD tools can add delay gates to avoid glitches and/or buffers that also strengthen signals.
    - Real gates with non-zero propagation delays can lead to a glitch
      * Sometimes exploited in timing circuits to produce a one-gate-wide phase 1 for the clock signal
  + Immediate logic expressions
    - Can help include intermediate columns in a truth table when implementing a complex logic expression
    - Can help to annotate intermediate gate outputs when analyzing a complex circuit diagram
  + Circuit Diagrams
    - Wire crossings
      * No circle/dot
        + Crossing of independent wires
      * Dot
        + Connected wires
    - An open circle means an inverter, or NOT gate.
      * Can be on gate inputs or outputs
    - No labels inside of the logic gate
      * Labels are used inside a drawn box for many components.
      * In later steps of abstraction, we will label inside of the drawn box.
  + Binary Numbers
    - Modulo arithmetic
      * Wrap around from 111…11 to 000…00 because of fixed number of bits
        + Carry out can be kept in separate status bit
      * Alternative is saturating arithmetic, which is useful in computer graphics with color values
      * NAND gates take fewer transistors to build than the AND gate itself.
* Timing Diagrams 11.5
  + Timing Diagram
    - Graphically shows a circuit’s output values for given input values that change over time.
    - If a = 0 and b = 0, then y = 0.
    - If a = 1 and b = 0, then y = 0.
    - If a = 0 and b = 1, then y = 0.
    - If a = 1 and b = 1, then y = 1.
  + Examples:
    - Some of the examples have an OR instead of an AND. Both of these can show up.
* Equations to/from circuits 11.6
  + Equations to Circuits
    - An equation is one way to represent a boolean function. Another way is a circuit.
    - You can convert an equation to a circuit by converting each operation to a gate.
      * Done first for items in the parentheses.
    - Converting behavior to a circuit is called **design.**
      * **Convert inside parentheses first.**
      * **Convert NOT before converting AND (or OR).**
      * **Convert AND.**
    - Converting a circuit to behavior (like an equation) is called analysis.
      * Start from inputs, replace gate by term.
      * Still at inputs, replace gate by term.
      * Moving towards output, convert gate to term, involving earlier terms.
      * Write the final equation.
    - A circuit whose output value is determined solely by the present combination of input values is called a **combinational circuit**
      * A combinational circuit is also called combinational logic.
    - A circuit whose output values may depend on the past sequence of input values, and not just the present input values, is called a **sequential circuit.**



* Basic properties of boolean algebra 11.8
  + The benefit of building circuits from logic gates, rather than directly from transistors, becomes clear after learning some basic properties of Boolean algebra.
  + More properties of boolean algebra.
* A summary of boolean algebra properties
* Sum of Products Form 11.9
  + Product Term
    - An anding of one or more variables like ab’c.
    - Sometimes called a product or a term.
    - An expression in sum of products form consists solely of an or-ing of product terms like ab’c + ab
  + A processor executes computer programs.
  + Various devices (like keyboards or USB ports) surrounding a processor may request the processor to execute a sub-program on behalf of that device, a request known as an **interrupt.**
  + Devices may be in two categories, low-priority and high-priority, and the processor may disable either category or both.
    - Low-priority: keyboard (k = 1), mouse (m = 1), USB port (u = 1). Disable all: p = 1.
    - High-priority: network interface (n = 1), battery backup (b = 1). Disable all: q = 1.
* Binary and Counting 11.10
  + Digital systems have two-valued signals (high, low) so digital systems use a base two number system.
  + A number in base ten is called a decimal number (from Latin "decem" meaning ten)
  + Base ten has ten symbols for a digit: 0, 1, ..., 9.
  + Base two has only two symbols for a digit: 0 and 1
    - Binary
  + 