

# **Rectangle Recognition Simplified**

## Four Lines That Do Not Form a Rectangle

---

---

---

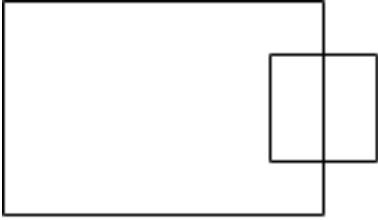
---

## Four Lines That Do Form a Rectangle



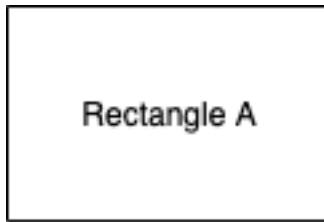
*Exercise – Write a pattern match that detects rectangles made up of straight line segments (e.g. PROLOG, or your favorite pattern matching language, or with loops or recursion in any general purpose language).*

## Smaller Rectangle Intersecting a Larger Rectangle



*Exercise - Write a pattern match that detects rectangles that intersect edges of bigger rectangles.*

## Rectangle With Text Inside It



*Exercise - Write a pattern match that detects all text items and associates them with rectangles.*

*(hint: create a bounding box for every text item)*

*Exercise - Write a pattern match that ensures that every text item is inside exactly one or zero rectangles.*

## Two Rectangles Connected by an Arrow



*Exercise: Write a pattern match that detects all arrows.*

*Exercise: Write a pattern match that determines the direction of every line.*

*Exercise: Write a pattern match that detects which rectangles are connected to other rectangles.*

## Annotated Arrow



*Exercise: Write a pattern match that detects all text items that are not inside rectangles.*

*Exercise: Write a pattern match that associates text items with each end of an arrow.*

*(hint: create bounding boxes for all text items)*

*Exercise: Convert lines to named input and output ports that are associated with rectangles.*

*Exercise: Write a match that ensures that every port has exactly one name.*

*Exercise: Write a match that ensures that every rectangle has exactly one text item in it and has exactly zero or one input port and exactly zero or one output port.*