

From Prose to Programs with Penrose

synthesizing domain-specific programs for diagram authoring with LLMs

from prose description...

“a parallelogram”



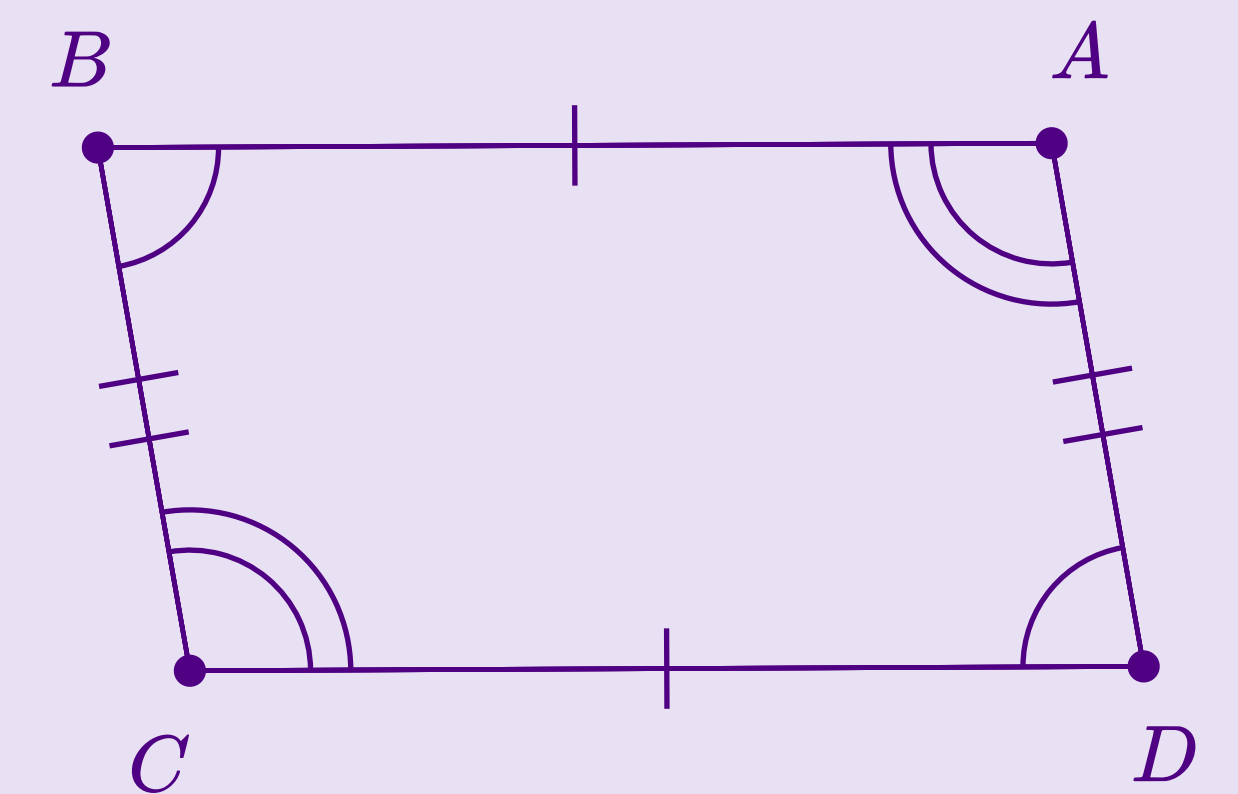
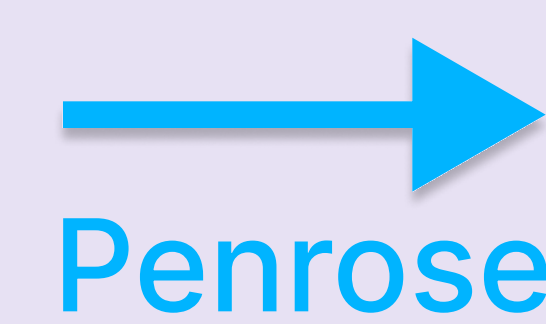
to DSL program...

```
Angle angleCDA := InteriorAngle(C, D, A)
Angle angleDAB := InteriorAngle(D, A, B)

-- Defining the Parallelogram
Quadrilateral parallelogramABCD :=
  Quadrilateral(A, B, C, D)
  Parallelogram(parallelogramABCD)

-- Marking the opposite angles equal
AngleMarker(angleABC, angleCDA)
```

to desired diagram!



anatomy of the LLM prompt

You are a code generator...
...Penrose is a system for authoring diagrams...

```
...
func1 ::= tname id "==" f1name "(" id ")"
pred1 ::= p1name "(" id ")"
...
p1name ::= "Parallelogram" // This predicate
makes a quadrilateral a parallelogram. Example
use: `...`
...
```

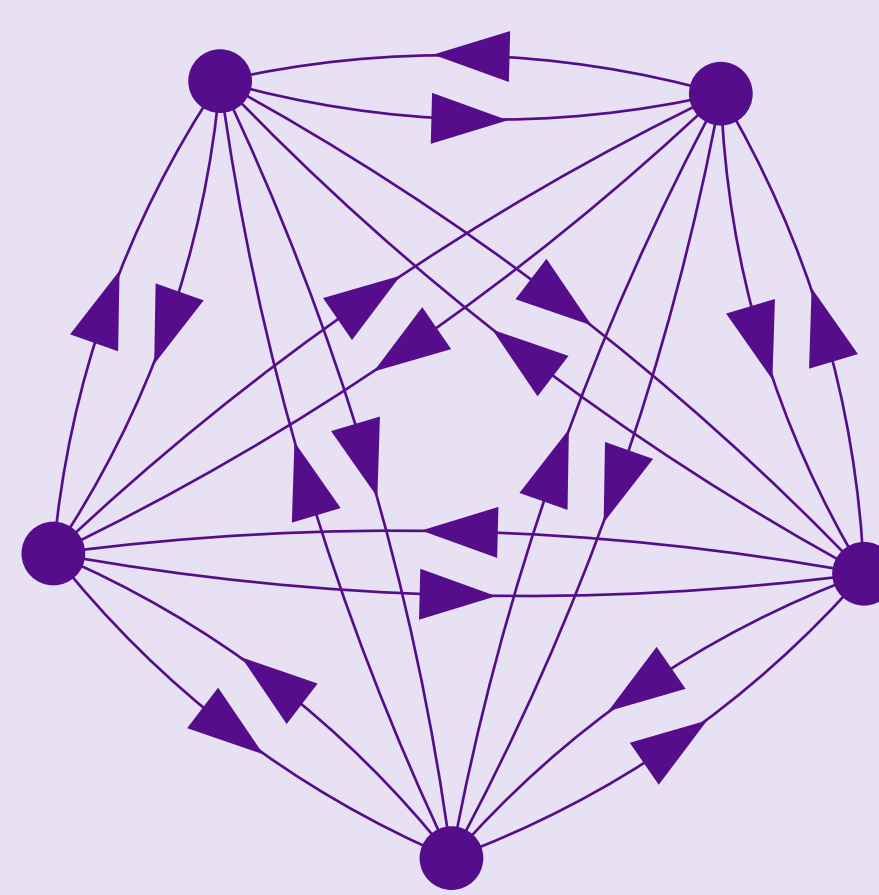
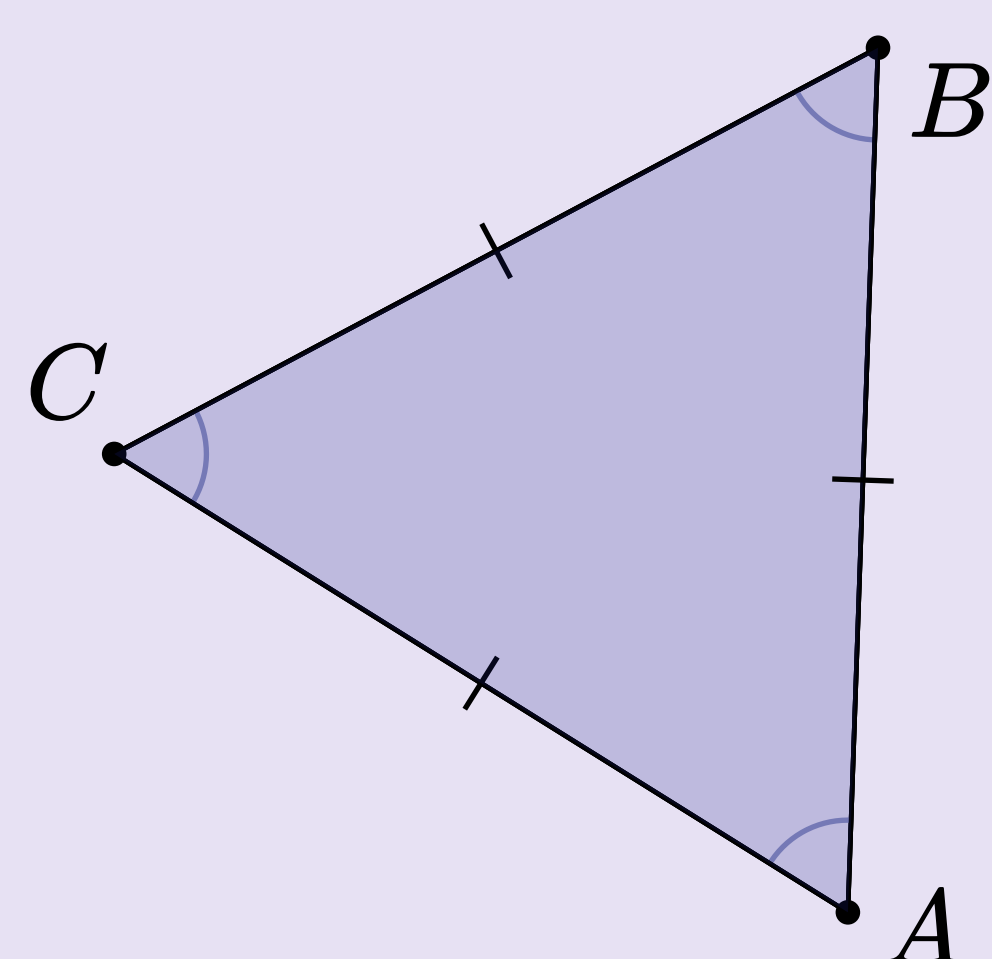
Draw a parallelogram ABCD. Draw the segments AB, BC, CD, and DA. Draw all four angles ABC, BCD...

use a **formal grammar** (72% avg. compile rate) over a schema language program (60%)

add **descriptive comments** (82% avg. compile) or sample programs (81% avg. compile)

write **thorough prose**, rather than terse prose (89% correct diagrams versus 70% correct)

for best results, combine these prompt techniques to generate programs: **96% compile w/o errors** where **80% are correct diagrams overall**.



LLMs take in text and return text... but pair them with **Penrose**, and now **anyone can make beautiful diagrams** with a few words.