APRiL

A geometric PRoof Language

What is APRiL

- Language: to write proofs in synthetic geometry
- Interpreter: automatic proof verification
- Library: to embed APRiL in applications
- Base library: commonly used definitions and statements to build own geometry courses

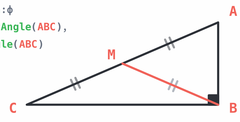
What makes APRiL special

- Accessible for non-professionals: the language is designed to be usable by people without expertise in formal proofs
- Multiplatform: supporting all major OS: Windows, macOS, Linux, iOS, Android
- Embeddable: can be used in constrained environments
- Opensource (soon): we are going to opensource APRiL as soon as it is stabilized

Who can use APRiL

- Teachers: for automating homework checking
- MOOC authors: for developing geometry courses and automating assignment verification
- Developers: for creating own APRiL applications
- Researchers: whose area of interest includes foundations of geometry

```
theorem RightTriangleMedian:ϕ
  is Triangle(ABC),is RightAngle(ABC),
  [BM] is Median for Triangle(ABC)
  ==> [AM] = [BM]
end
```




```
proof theorem RightTriangleMedian:
let Line(a) be Perpendicular for (Line(AB), A).
let Line(b) be Perpendicular for (Line(BC), C).
let D be IntersectionPoint for (Line(a), Line(b)).
is Rectangle(ABCD).
let E be IntersectionPoint for ([AC], [BD]).
[AC] = [BD] by theorem RectangleDiagonals(is Rectangle(ABCD)).
[AE] = [BE].
M == E.
[AM] = [BM].
end
```

APRiL: primitives

Points	A, B, A’, Z12
Segments	[AB], [s]
Lines	Line(AB), Line(l)
Rays	Ray(AB), Ray(r)
Circles	Circle(OA), Circle(O, [AB]), Circle(o)
Angles	Angle(ABC), Angle(a)
Halfplanes	Halfplane(Line(l), P), Halfplane(h)

APRiL: predefined relations

Between	[ABC]
Coincidence	A == B
Incidence	P in Circle(o)
Congruence	Halfplane(h1) = Halfplane(h2)

APRiL: predefined functions

Distance	|AB|
Angle measure	angle(ABC)
Area	area(ABC)

APRiL: user-defined notions

- Relations are used to define “relations” between geometric objects: Collinear, Parallel, Intersected
- Objects are used to define new geometric figures: Triangle, Rectangle, Trapezoid
- Definitions are used to define geometric relations between objects: Median, Bisector, Circumcircle

APRiL: statements

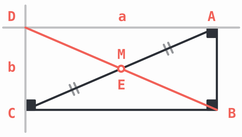
```
axiom TriangleInequality:
==>
|AB| + |BC| >= |AC|
end
```

```
theorem SAS:
[AB] = [A’B’],
Angle(ABC) = Angle(A’B’C’),
[BC] = [B’C’]
==>
Triangle(ABC) = Triangle(A’B’C’)
End
```

APRiL: proofs

Proof in APRiL is a sequence of proof steps:

```
proof theorem RightTriangleMedian:
let Line (a) be Perpendicular for (Line(AB), A).
let Line (b) be Perpendicular for (Line(BC), C).
let D be IntersectionPoint for(Line(a), Line(b)).
is Rectangle(ABCD).
let E be IntersectionPoint for ([AC], [BD]).
[AC] = [BD] by theorem RectangleDiagonals(is Rectangle(ABCD)).
[AE] = [BE].
M == E.
[AM] = [BM].
end
```



APRiL: proof steps

There are the following proof step types:

- **Assert:** claims the truth of a logical expression
- **Construct:** performs a geometric construction
- **Select point:** selects a new point on the plane or on a primitive
- **Contradiction:** claims that a contradiction was reached in the previous step

APRiL: justifications

To justify a proof step, you can use justifications:

- **Empty:** instructs the verifier to automatically find proof of the step
- **Statement application:** provides the verifier with a statement that can justify the step
- **Full:** the separate proof of validity of current step
- **Assume:** assumes truth of a logical expression. It helps to prove implications or is used in proof by contradiction
- **Case:** used in proofs by cases
- **Similar to:** instructs the verifier to find a proof by analogy with an already proved step