Euclidean Logic

October 7, 2022

The First Proposition of Euclid

We consider a formalisation of just enough Euclidean Geometry to prove the first proposition of Euclid. This formalisation sits on top of a fragment of Kelley-Morse set theory. We illustrate the axioms, definitions and proof using a proof assistant we are developing which is based on linearised natural deduction for classical and intuitionistic first-order logic with equality and an extension-operator.

Here are the axioms and definitions (we do not show the Kelley-Morse axioms and definitions):

```
8. Set(P) & Set(M)
9. Line(l,x,y) -> ((l ⊂ P) & ((x ϵ P) & ((y ϵ P) & ¬(x = y))))
10. ((x ϵ P) & ((y ϵ P) & ¬(x = y))) -> ∃l.Line(l,x,y)
11. ((x ϵ P) & (r ϵ M)) -> ∃c.Circle(c,x,r)
12. (Circle(c,x,r) & (y ϵ c)) -> (¬(y = x) & ((d'(y,x)) = r))
13. Circle(c,x,r) -> ((c ⊂ P) & ((x ϵ P) & (r ϵ M)))
14. ((x ϵ P) & ((y ϵ P) & ¬(x = y))) -> ((d'(x,y)) ϵ M)
15. (Circle(c1,x,(d'(x,y))) & Circle(c2,y,(d'(x,y)))) -> ∃z.(z ϵ (c1 ∩ c2))
16. (P ∩ M) = 0
17. (d'(x,y)) = (d'(y,x))
18. Line(l,x,y) <-> Line(l,y,x)

Triangle(l1,l2,l3,x,y,z) <-> (Line(l1,x,y) & (Line(l2,y,z) & Line(l3,z,x)))
EquiLat(l1,l2,l3,x,y,z) <-> (Triangle(l1,l2,l3,x,y,z) &
(((d'(x,y)) = (d'(y,z))) & ((d'(y,z)) = (d'(z,x)))))
```

Here is the proof of the first proposition. We interpret Euclidean 'construction' as an intuitionistic existential quantifier.

```
0. Line(l,x,y)  Hyp
1. Line(l,x,y) -> ((l ⊂ P) & ((x ϵ P) & ((y ϵ P) & ¬(x = y))))  AxInt
2. (l ⊂ P) & ((x ϵ P) & ((y ϵ P) & ¬(x = y)))  ImpElim 0 1
3. (x ϵ P) & ((y ϵ P) & ¬(x = y))  AndElimR 2
4. ((x ϵ P) & ((y ϵ P) & ¬(x = y))) -> ((d'(x,y)) ϵ M)  AxInt
5. (d'(x,y)) ϵ M  ImpElim 3 4
6. x ϵ P  AndElimL 3
7. (x ϵ P) & ((d'(x,y)) ϵ M)  AndInt 6 5
8. ((x ϵ P) & (r ϵ M)) -> ∃c.Circle(c,x,r)  AxInt
9. ∀r.(((x ϵ P) & (r ϵ M)) -> ∃c.Circle(c,x,r))  ForallInt 8
```

```
10. ((x ϵ P) & ((d'(x,y)) ϵ M)) -> ∃c.Circle(c,x,(d'(x,y)))  ForallElim 9
11. ∃c.Circle(c,x,(d'(x,y)))  ImpElim 7 10
12. (y ϵ P) & ¬(x = y)  AndElimR 3
13. y ϵ P  AndElimL 12
14. (y ϵ P) & ((d'(x,y)) ϵ M)  AndInt 13 5
15. ∀x.(((x ϵ P) & (r ϵ M)) -> ∃c.Circle(c,x,r))  ForallInt 8
16. ((y ϵ P) & (r ϵ M)) -> ∃c.Circle(c,y,r)  ForallElim 15
17. ∀r.(((y ϵ P) & (r ϵ M)) -> ∃c.Circle(c,y,r))  ForallInt 16
18. ((y ϵ P) & ((d'(x,y)) ϵ M)) -> ∃c.Circle(c,y,(d'(x,y)))  ForallElim 17
19. ∃c.Circle(c,y,(d'(x,y)))  ImpElim 14 18
20. Circle(c1,x,(d'(x,y)))  Hyp
21. Circle(c2,y,(d'(x,y)))  Hyp
22. Circle(c1,x,(d'(x,y))) & Circle(c2,y,(d'(x,y)))  AndInt 20 21
23. (Circle(c1,x,(d'(x,y))) & Circle(c2,y,(d'(x,y)))) -> ∃z.(z ϵ (c1 ∩ c2))  AxInt
24. ∃z.(z ϵ (c1 ∩ c2))  ImpElim 22 23
25. z ϵ (c1 ∩ c2)  Hyp
26. ((z ϵ (x ∪ y)) <-> ((z ϵ x) v (z ϵ y))) & ((z ϵ (x ∩ y)) <->
((z ϵ x) & (z ϵ y)))  TheoremInt
27. (z ϵ (x ∩ y)) <-> ((z ϵ x) & (z ϵ y))  AndElimR 26
28. ((z ϵ (x ∩ y)) -> ((z ϵ x) & (z ϵ y))) & (((z ϵ x) & (z ϵ y))
-> (z ϵ (x ∩ y)))  EquivExp 27
29. (z ϵ (x ∩ y)) -> ((z ϵ x) & (z ϵ y))  AndElimL 28
30. ∀x.((z ϵ (x ∩ y)) -> ((z ϵ x) & (z ϵ y)))  ForallInt 29
31. (z ϵ (c1 ∩ y)) -> ((z ϵ c1) & (z ϵ y))  ForallElim 30
32. ∀y.((z ϵ (c1 ∩ y)) -> ((z ϵ c1) & (z ϵ y)))  ForallInt 31
33. (z ϵ (c1 ∩ c2)) -> ((z ϵ c1) & (z ϵ c2))  ForallElim 32
34. (z ϵ c1) & (z ϵ c2)  ImpElim 25 33
35. z ϵ c1  AndElimL 34
36. z ϵ c2  AndElimR 34
37. Circle(c,x,r) -> ((c ⊂ P) & ((x ϵ P) & (r ϵ M)))  AxInt
38. ∀c.(Circle(c,x,r) -> ((c ⊂ P) & ((x ϵ P) & (r ϵ M))))  ForallInt 37
39. Circle(c1,x,r) -> ((c1 ⊂ P) & ((x ϵ P) & (r ϵ M)))  ForallElim 38
40. ∀r.(Circle(c1,x,r) -> ((c1 ⊂ P) & ((x ϵ P) & (r ϵ M))))  ForallInt 39
41. Circle(c1,x,(d'(x,y))) -> ((c1 ⊂ P) & ((x ϵ P) & ((d'(x,y)) ϵ M)))
 ForallElim 40
42. (c1 ⊂ P) & ((x ϵ P) & ((d'(x,y)) ϵ M))  ImpElim 20 41
43. c1 ⊂ P  AndElimL 42
44. ∀z.((z ϵ c1) -> (z ϵ P))  DefExp 43
45. (z ϵ c1) -> (z ϵ P)  ForallElim 44
46. z ϵ P  ImpElim 35 45
47. (Circle(c,x,r) & (y ϵ c)) -> (¬(y = x) & ((d'(y,x)) = r))  AxInt
48. ∀c.((Circle(c,x,r) & (y ϵ c)) -> (¬(y = x) & ((d'(y,x)) = r)))  ForallInt 47
49. (Circle(c1,x,r) & (y ϵ c1)) -> (¬(y = x) & ((d'(y,x)) = r))  ForallElim 48
50. ∀y.((Circle(c1,x,r) & (y ϵ c1)) -> (¬(y = x) & ((d'(y,x)) = r)))  ForallInt 49
51. (Circle(c1,x,r) & (z ϵ c1)) -> (¬(z = x) & ((d'(z,x)) = r))  ForallElim 50
52. ∀r.((Circle(c1,x,r) & (z ϵ c1)) -> (¬(z = x) & ((d'(z,x)) = r)))  ForallInt 51
53. (Circle(c1,x,(d'(x,y))) & (z ϵ c1)) -> (¬(z = x)
& ((d'(z,x)) = (d'(x,y))))  ForallElim 52
54. Circle(c1,x,(d'(x,y))) & (z ϵ c1)  AndInt 20 35
55. ¬(z = x) & ((d'(z,x)) = (d'(x,y)))  ImpElim 54 53
```

```
56. ¬(z = x)  AndElimL 55
57. (x ϵ P) & ¬(z = x)  AndInt 6 56
58. (z ϵ P) & ((x ϵ P) & ¬(z = x))  AndInt 46 57
59. ((x ϵ P) & ((y ϵ P) & ¬(x = y))) -> ∃l.Line(l,x,y)  AxInt
60. ∀x.(((x ϵ P) & ((y ϵ P) & ¬(x = y))) -> ∃l.Line(l,x,y))  ForallInt 59
61. ((z ϵ P) & ((y ϵ P) & ¬(z = y))) -> ∃l.Line(l,z,y)  ForallElim 60
62. ∀y.(((z ϵ P) & ((y ϵ P) & ¬(z = y))) -> ∃l.Line(l,z,y))  ForallInt 61
63. ((z ϵ P) & ((x ϵ P) & ¬(z = x))) -> ∃l.Line(l,z,x)  ForallElim 62
64. ∃l.Line(l,z,x)  ImpElim 58 63
65. Line(l1,z,x)  Hyp
66. ∀y.((Circle(c,x,r) & (y ϵ c)) -> (¬(y = x) & ((d'(y,x)) = r)))  ForallInt 47
67. (Circle(c,x,r) & (z ϵ c)) -> (¬(z = x) & ((d'(z,x)) = r))  ForallElim 66
68. ∀x.((Circle(c,x,r) & (z ϵ c)) -> (¬(z = x) & ((d'(z,x)) = r)))  ForallInt 67
69. (Circle(c,y,r) & (z ϵ c)) -> (¬(z = y) & ((d'(z,y)) = r))  ForallElim 68
70. ∀c.((Circle(c,y,r) & (z ϵ c)) -> (¬(z = y) & ((d'(z,y)) = r)))  ForallInt 69
71. (Circle(c2,y,r) & (z ϵ c2)) -> (¬(z = y) & ((d'(z,y)) = r))  ForallElim 70
72. Circle(c2,y,(d'(x,y))) & (z ϵ c2)  AndInt 21 36
73. ∀r.((Circle(c2,y,r) & (z ϵ c2)) -> (¬(z = y) & ((d'(z,y)) = r)))  ForallInt 71
74. (Circle(c2,y,(d'(x,y))) & (z ϵ c2)) -> (¬(z = y)
 & ((d'(z,y)) = (d'(x,y))))  ForallElim 73
75. ¬(z = y) & ((d'(z,y)) = (d'(x,y)))  ImpElim 72 74
76. ¬(z = y)  AndElimL 75
77. (y ϵ P) & ¬(z = y)  AndInt 13 76
78. (z ϵ P) & ((y ϵ P) & ¬(z = y))  AndInt 46 77
79. ∀x.(((x ϵ P) & ((y ϵ P) & ¬(x = y))) -> ∃l.Line(l,x,y))  ForallInt 59
80. ((z ϵ P) & ((y ϵ P) & ¬(z = y))) -> ∃l.Line(l,z,y)  ForallElim 79
81. ∃l.Line(l,z,y)  ImpElim 78 80
82. Line(l2,z,y)  Hyp
83. Line(l,x,y) <-> Line(l,y,x)  AxInt
84. ∀x.(Line(l,x,y) <-> Line(l,y,x))  ForallInt 83
85. Line(l,z,y) <-> Line(l,y,z)  ForallElim 84
86. ∀l.(Line(l,z,y) <-> Line(l,y,z))  ForallInt 85
87. Line(l2,z,y) <-> Line(l2,y,z)  ForallElim 86
88. (Line(l2,z,y) -> Line(l2,y,z)) & (Line(l2,y,z) -> Line(l2,z,y))  EquivExp 87
89. Line(l2,z,y) -> Line(l2,y,z)  AndElimL 88
90. Line(l2,y,z)  ImpElim 82 89
91. Line(l2,y,z) & Line(l1,z,x)  AndInt 90 65
92. Line(l,x,y) & (Line(l2,y,z) & Line(l1,z,x))  AndInt 0 91
93. Triangle(l,l2,l1,x,y,z)  DefSub 92
94. (d'(z,x)) = (d'(x,y))  AndElimR 55
95. (d'(z,y)) = (d'(x,y))  AndElimR 75
96. (d'(x,y)) = (d'(z,y))  Symmetry 95
97. (d'(z,x)) = (d'(z,y))  EqualitySub 94 96
98. (d'(x,y)) = (d'(y,x))  AxInt  Qed
99. ∀x.((d'(x,y)) = (d'(y,x)))  ForallInt 98
100. (d'(z,y)) = (d'(y,z))  ForallElim 99
101. (d'(x,y)) = (d'(y,z))  EqualitySub 96 100
102. (d'(z,x)) = (d'(y,z))  EqualitySub 97 100
103. (d'(y,z)) = (d'(z,x))  Symmetry 102
104. ((d'(x,y)) = (d'(y,z))) & ((d'(y,z)) = (d'(z,x)))  AndInt 101 103
```

```
105. Triangle(l,l2,l1,x,y,z) & (((d'(x,y)) = (d'(y,z)))
& ((d'(y,z)) = (d'(z,x))))  AndInt 93 104
106. EquiLat(l,l2,l1,x,y,z)  DefSub 105
107. ∃z.EquiLat(l,l2,l1,x,y,z)  ExistsInt 106
108. ∃l1.∃z.EquiLat(l,l2,l1,x,y,z)  ExistsInt 107
109. ∃l2.∃l1.∃z.EquiLat(l,l2,l1,x,y,z)  ExistsInt 108
110. ∃l2.∃l1.∃z.EquiLat(l,l2,l1,x,y,z)  ExistsElim 81 82 109
111. ∃l2.∃l1.∃z.EquiLat(l,l2,l1,x,y,z)  ExistsElim 64 65 110
112. ∃l2.∃l1.∃z.EquiLat(l,l2,l1,x,y,z)  ExistsElim 24 25 111
113. ∃l2.∃l1.∃z.EquiLat(l,l2,l1,x,y,z)  ExistsElim 19 21 112
114. ∃l2.∃l1.∃z.EquiLat(l,l2,l1,x,y,z)  ExistsElim 11 20 113
115. Line(l,x,y) -> ∃l2.∃l1.∃z.EquiLat(l,l2,l1,x,y,z)  ImpInt 114 Qed
```