Projekt Algebra:

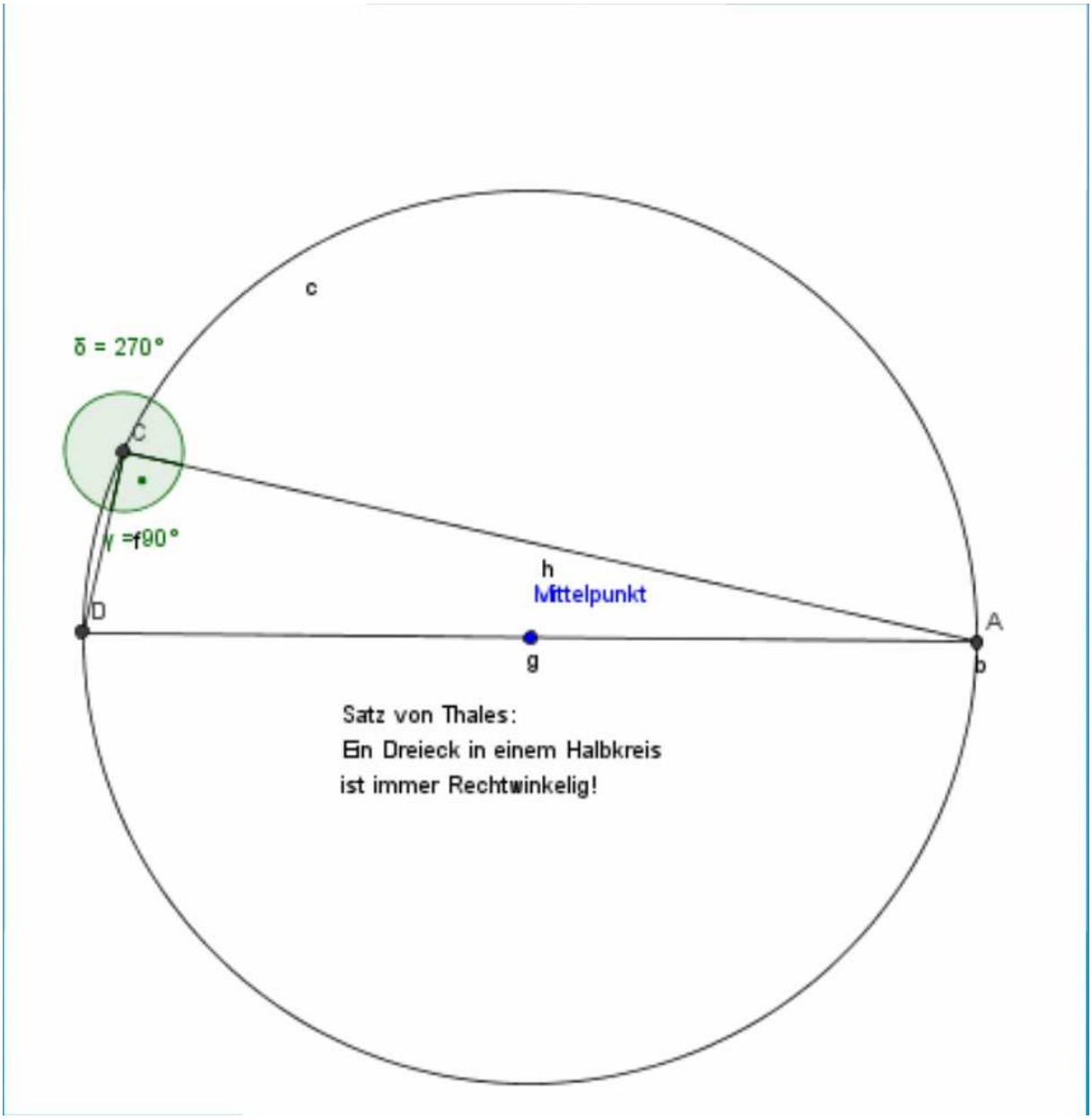
Automatisches Beweisen in der Geometrie

geleitet von:
- Erhard Aichinger
- Zoltan Kovacs

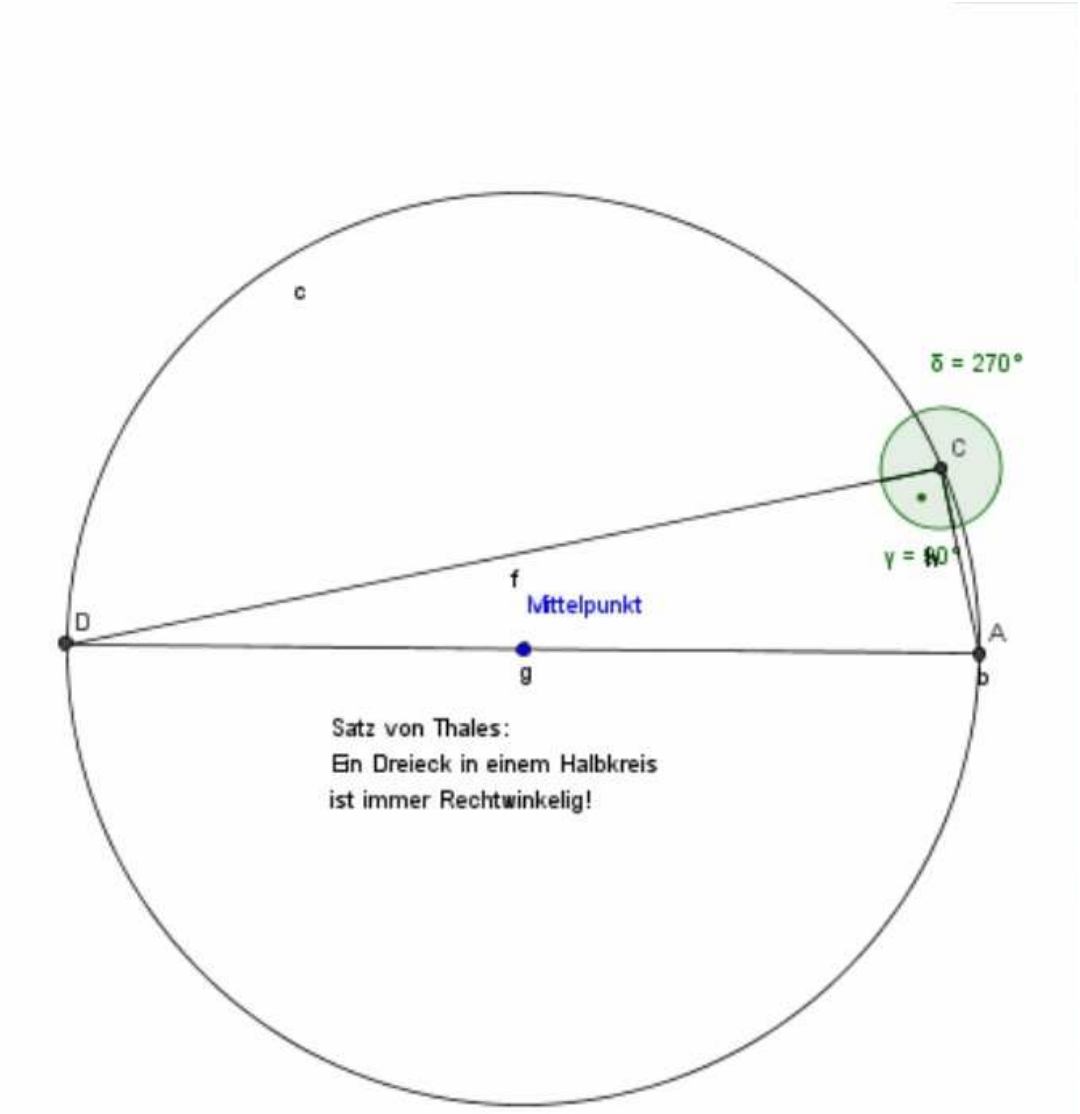
WAS HABEN WIR GEMACHT?

Ziel: Beweisen von geometrischen Sätzen

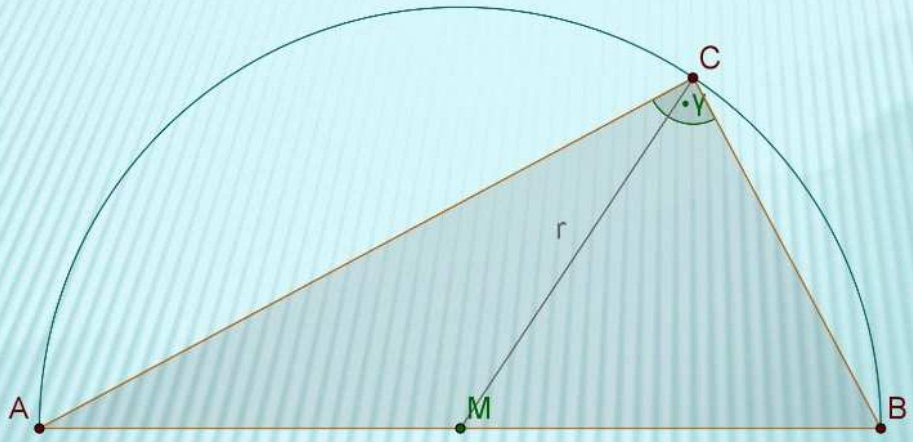
- Grundidee:
  Voraussetzungen → Konklusion
- Ablauf:
  - Theorie
  - Sage
  - Beweisen
- Problem: Geometrie - Algebra


c
δ = 270°
C
γ = 90°
h
Mittelpunkt
D
A
g
b
Satz von Thales:
Ein Dreieck in einem Halbkreis
ist immer Rechtwinkelig!


δ = 270°
C
c
γ = 90°
f
h
Mittelpunkt
D
A
g
b
Satz von Thales:
Ein Dreieck in einem Halbkreis
ist immer Rechtwinkelig!


c
δ = 270°
C
γ = 90°
f
Mittelpunkt
D
A
g
b
Satz von Thales:
Ein Dreieck in einem Halbkreis
ist immer Rechtwinkelig!

GLEICHUNGSSYSTEM



$$M = \frac{A + B}{2}$$

$$AM = CM$$

$$\rightarrow \gamma = 90°$$

GLEICHUNGSSYSTEM

```
def Mp(A,B,C):
    return 2*C[0]-A[0]-B[0], 2*C[1]-A[1]-B[1]

def Aq(A,B):
    return (A[0]-B[0])^2+(A[1]-B[1])^2

def sr(A,B,C,D):
    return (A[0]-B[0])*(C[0]-D[0])+(A[1]-B[1])*(C[1]-D[1])


Loesung([Mp(A,B,M), Aq(A,M)-Aq(C,M), sr(A,C,B,C)*z-1])
```

Lösung ➔ Gegenbeispiel gefunden ➔ gilt nicht

Keine Lösung ➔ kein Gegenbeispiel vorhanden ➔ Beweis gelungen, Satz gilt

GEOMETRISCHER SATZ → ALGEBRAISCHE AUSSAGE

Verwendung von Koordinatengeometrie

z.B: $A=(a_1|a_2)$

- M ist Mittelpunkt von AB

  Formel: $2M=A+B$

→ 2 Gleichungen:

$2m_1 - a_1 - b_1 = 0$ (x-Koordinaten)

$2m_2 - a_2 - b_2 = 0$ (y-Koordinaten)

GEOMETRISCHER SATZ → ALGEBRAISCHE AUSSAGE

- Winkel = 90°

→Skalarprodukt

$(a_1 - c_1)(b_1 - c_1) + (a_2 - c_2)(b_2 - c_2) = 0$

Selbe Vorgehensweise mit anderen geometrischen Aussagen

Satz von Thales

52 minutes ago by Felix

```
def Variablen(v):
    global Ring
    Ring=PolynomialRing(QQ,v,order='dp')
    var(v)
def Loesung(Gs):
    Gs=flatten([Gs])
    I=Gs*Ring
    if (I.groebner_basis()==[1]):
        print "NEIN - Das Gleichungssystem hat keine Lösung."
    else:
        print "JA - Das Gleichungssystem hat eine Lösung."
        print I.interreduced_basis()
def Mp(A,B,C):
    return 2*C[0]-A[0]-B[0], 2*C[1]-A[1]-B[1]
    #Mittelpunkt
def Aq (A,B):
    return (A[0]-B[0])^2+(A[1]-B[1])^2
    #Abstand (im Quadrat)
def sr(A,B,C,D):
    return (A[0]-B[0])*(C[0]-D[0])+(A[1]-B[1])*(C[1]-D[1])
    #Senkrechte (AB auf CD)
def Coll(A,B,C):
    return C[0]*A[1]-C[0]*B[1]+C[1]*B[0]-A[0]*C[1]+A[0]*B[1]-B[0]*A[1]
    #Kollinear
```

```
Variablen('a1,a2,b1,b2,c1,c2,m1,m2,z')

A=(a1,a2)
B=(b1,b2)
C=(c1,c2)
M=(m1,m2)

p1=Mp(A,B,M)
p2=Aq(A,M)-Aq(C,M)

q=sr(A,C,B,C)*z-1

Loesung([p1,p2,q])
```

```
NEIN - Das Gleichungssystem hat keine Lösung.
```

```
Loesung([Mp(A,B,F),Mp(A,C,E),Coll(F,S,C),Coll(E,S,B),(Aq(F,S)*4-Aq(S,C))*z1-1])
```

```
Das Gleichungssystem hat eine Lösung.
[c1^2*z1 + c2^2*z1 - 4*f1^2*z1 - 4*f2^2*z1 - 2*c1*s1*z1 + 8*f1*s1*z1 -
3*s1^2*z1 - 2*c2*s2*z1 + 8*f2*s2*z1 - 3*s2^2*z1 + 1, c2*e1 - c1*e2 -
2*e2*f1 + 2*e1*f2 - c2*s1 + 3*e2*s1 - 2*f2*s1 + c1*s2 - 3*e1*s2 +
2*f1*s2, c2*f1 - c1*f2 - c2*s1 + f2*s1 + c1*s2 - f1*s2, a1 + c1 - 2*e1,
a2 + c2 - 2*e2, b1 - c1 + 2*e1 - 2*f1, b2 - c2 + 2*e2 - 2*f2]
```

```
Loesung([Mp(A,B,F),Mp(A,C,E),Coll(F,S,C),Coll(E,S,B),(Aq(F,S)*4-Aq(S,C))*z1-1,Coll(A,B,C)*z2-1])
```

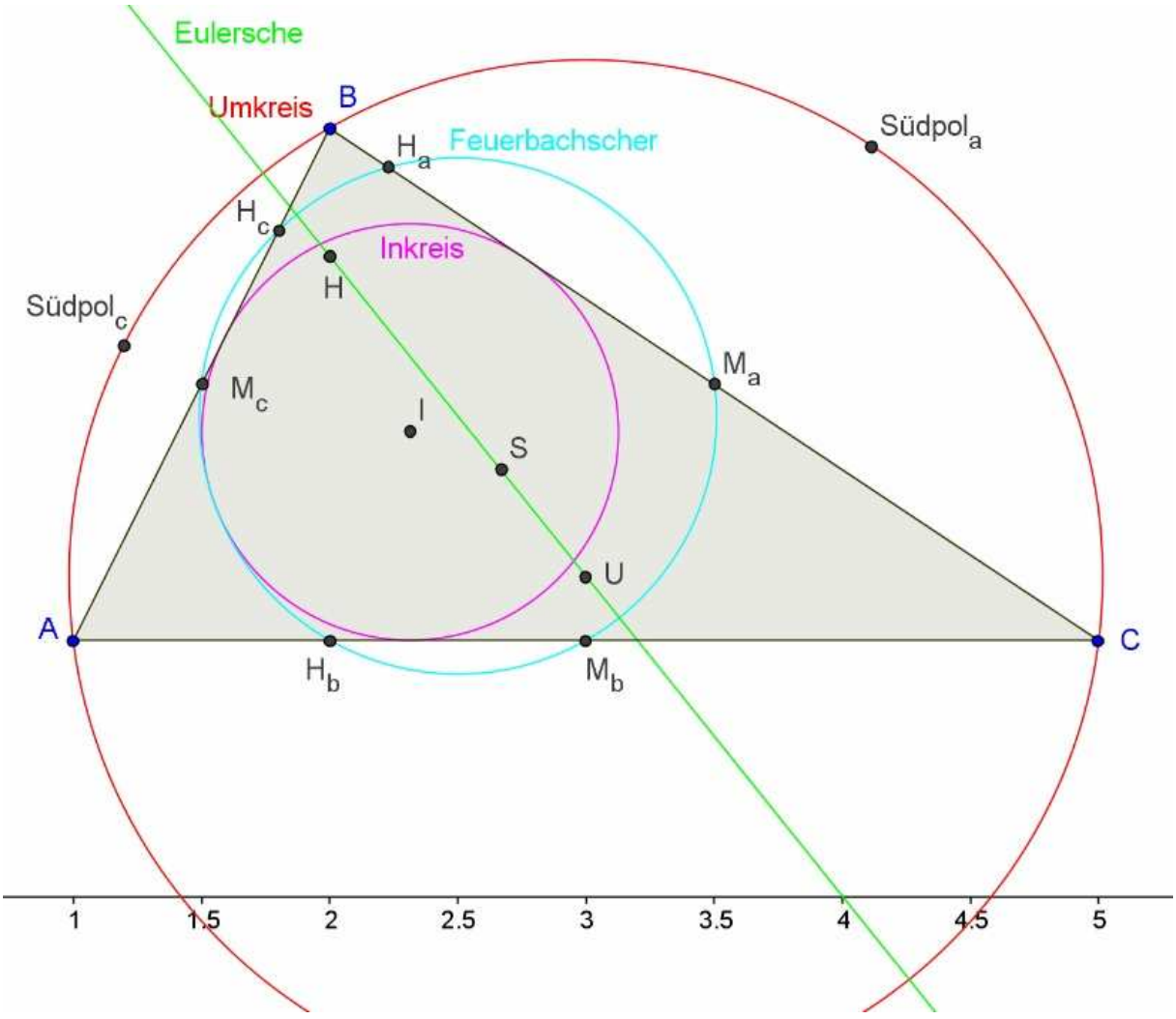
```
Das Gleichungssystem hat keine Lösung.
```

```
Loesung([p1,p2,p4,p5,p7,p8,p9,p10,not1,q])
```

```
^C
Traceback (most recent call last):        I=Gs*Ring
  File "", line 1, in <module>

  File "/tmp/tmpJXgb7J/___code___.py", line 59, in <module>
    exec compile(u'Loesung([p1,p2,p4,p5,p7,p8,p9,p10,not1,q])
  File "", line 1, in <module>

  File "/tmp/tmpJXgb7J/___code___.py", line 10, in Loesung
    if (I.groebner_basis()==[_sage_const_1 ]):
  File "/opt/sage-4.8/local/lib/python2.6/site-packages/sage/misc/cachefunc.py", line 555, in __call__
    w = self._cachedmethod._instance_call(self._instance, *args, **kwds)
  File "/opt/sage-4.8/local/lib/python2.6/site-packages/sage/misc/cachefunc.py", line 778, in _instance_call
    return self._cachedfunc.f(inst, *args, **kwds)
  File "/opt/sage-4.8/local/lib/python2.6/site-packages/sage/rings/polynomial/multi_polynomial_ideal.py", line 3158, in groebne
    gb = self._groebner_basis_libsingular("groebner", deg_bound=deg_bound, mult_bound=mult_bound, *args, **kwds)
  File "/opt/sage-4.8/local/lib/python2.6/site-packages/sage/rings/polynomial/multi_polynomial_ideal.py", line 502, in wrapper
    return func(*args, **kwds)
  File "/opt/sage-4.8/local/lib/python2.6/site-packages/sage/rings/polynomial/multi_polynomial_ideal.py", line 1600, in _groebn
    S = groebner(self)
  File "function.pyx", line 1036, in sage.libs.singular.function.SingularFunction.__call__ (sage/libs/singular/function.cpp:101
  File "function.pyx", line 1204, in sage.libs.singular.function.call_function (sage/libs/singular/function.cpp:11392)
```



Erhard

Zoltan

DANKE FÜR IHRE AUFMERKSAMKEIT !

Donau Moritz, Felhofer Emil, Ganglbauer Johanna, Gruber Sonja, Heinzelreiter Bernhard, Hinterreither Armin, Hultsch Klara, Mittag Felix, Moosbauer Jakob, Stellner Ricarda, Weidenholzer Lukas

presentation by Felix Mittag