Quantifier Elimination in Maple 2023

Description

- The QuantifierElimination package has been added in Maple 2023. This package offers a new set of routines for quantifier elimination over the reals (QE). This package is the first package included with Maple to offer Virtual Term Substitution (VTS) as an algorithm for QE. Furthermore it offers a new implementation of Cylindrical Algebraic Decomposition (CAD) being the first implementation using the Lazard projection with equational constraints as part of CAD both for QE and standalone CADs to explore real algebraic geometry. Lastly, a new poly-algorithm offers a new way to use VTS in conjunction with CAD for QE including in incremental contexts.
- QuantifierEliminate is the standard routine for elimination:

```
> with(QuantifierElimination);
```

$$[CylindricalAlgebraicDecompose, DeleteFormula, InsertFormula, PartialCylindricalAlgebraicDecompose, QuantifierEliminate, QuantifierTools]$$

```
> expr := exists( x, forall( y, exists( z, And(4*x^2 + x*y^2 - z + 1/4
  = 0, 2*x + y^2*z + 1/2 = 0, x^2*z - 1/2*x - y^2 = 0 ) ) ) );
```

$$expr := \exists\left(x, \forall\left(y, \exists\left(z, 4x^2 + xy^2 - z + \frac{1}{4} = 0 \wedge 2x + y^2 z + \frac{1}{2} = 0 \wedge x^2 z - \frac{1}{2}x - y^2 = 0\right)\right)\right)$$

```
> QuantifierEliminate(expr);
```

$$false$$

- You can opt to include with the result witnesses that prove equivalence of quantifier-free formulae to a quantified expression:

```
> expr := exists([v__1, v__2, v__3, v__4, v__5, v__6, v__7, v__8],And
  (v__7 < 0,0 < v__8,0 < v__4,v__2*v__6+v__3*v__8 = v__4,v__1*v__5+
  v__3*v__7 = v__4,v__6 = 1,v__5 = 1, 0 < v__1));
```

$$expr := \exists\left([v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8], v_7 < 0 \wedge 0 < v_8 \wedge 0 < v_4 \wedge v_2 v_6 + v_3 v_8 = v_4 \wedge v_1 v_5 + v_3 v_7 = v_4 \wedge v_6 = 1 \wedge v_5 = 1 \wedge 0 < v_1\right)$$

```
> (qf, witnesses) := QuantifierEliminate( expr, 'output = [ qf,
  witnesses]' );
```

$$qf, witnesses := true, \left[\left[true, v_8 = \frac{9}{2}, v_7 = -\frac{1}{2}, v_6 = 1, v_5 = 1, v_4 = \frac{1}{4}, v_3 = \frac{1}{2}, v_2 = -2, v_1 = \frac{1}{2}\right]\right]$$

```
> map(evalb,eval(op(2,expr),witnesses[1][2..-1]));
```

$$true \wedge true \wedge true \wedge true \wedge true \wedge true \wedge true$$

- PartialCylindricalAlgebraicDecompose offers QE purely via partial CAD:

```
> expr := exists(c,forall([b, a],Implies(Or(And(a-d = 0,b-c = 0),And
  (a-c = 0,b-1 = 0)),a^2-b = 0)));
```

$$expr := \exists\left(c, \forall\left([b, a], (a - d = 0 \wedge b - c = 0) \vee (a - c = 0 \wedge b - 1 = 0) \quad a^2 - b = 0\right)\right)$$

```
> PartialCylindricalAlgebraicDecompose(expr);
```

$$d = -1 \vee d = 1$$

- QuantifierTools is a subpackage of QuantifierElimination that offers tools for working with and manipulating Tarski formulae:

```
> with(QuantifierTools);
```

$$[AlphaConvert, ConvertRationalConstraintsToTarski, ConvertToPrenexForm, GetAllPolynomials, GetEquationalConstraints, GetUnquantifiedFormula, NegateFormula, SuggestCADOptions]$$

```
> uqf := GetUnquantifiedFormula( expr );
```

$$uqf := (a - d = 0 \wedge b - c = 0) \vee (a - c = 0 \wedge b - 1 = 0) \quad a^2 - b = 0$$

- CylindricalAlgebraicDecompose offers production of CADData objects that can be inspected to explore a cylindrical algebraic decomposition:

```
> C := CylindricalAlgebraicDecompose( GetUnquantifiedFormula( expr ),
  'variablestrategy' = [ d, c, b, a ] );
```

$$C := CADData\ for\ set\ of\ polynomials\ in\ \{d, c, b, a\}$$

```
> leaves := GetLeafCells( C )[ 1 .. 5 ];
```

$$leaves :=$$

$[$*Level 4 CADCell with local description RootOf(_Z^2-b,index = real[2]) < a and local sample point a = 7,*

Level 4 CADCell with local description a = RootOf(_Z^2-b,index = real[2]) and local sample point a = RootOf(_Z^2-26,481588776929776967857 85/9444732965739290427392 .. 2407943884648884839293 3/4722366482869645213696),

Level 4 CADCell with local description And(c < a,a < RootOf(_Z^2-b,index = real[2])) and local sample point a = 61/12, Level 4 CADCell with local description a = c and local sample point a = 5,

Level 4 CADCell with local description And(d < a,a < c) and local sample point a = 3$]$

```
> cell := leaves[1];
```

$$cell :=$$

Level 4 CADCell with local description RootOf(_Z^2-b,index = real[2]) < a and local sample point a = 7

```
> GetFullDescription(cell);
```

$$1 < d \wedge d^2 < c \wedge c^2 < b \wedge RootOf\left(\_Z^2 - b, index = real_2\right) < a$$

```
> GetSamplePoint(cell);
```

$$[d = 2, c = 5, b = 26, a = 7]$$

```
> SignOfPolynomialOnCell( C, c-a, cell );
```

$$-1$$

```
> SignOfPolynomialOnCell( C, d-1, cell );
```

$$1$$