Caspy - A Python CAS

Caspy is a CAS that has been developed using Lark parser and Python. Some of the key feautures are:

- Factorisation of univariate polynomials
- Integration and differentiation of symbollic variables
- Expansion of trigonometric expressions, eg expand sin(2x) as 2sin(x)cos(x)
- Expansion of expressions in brackets
- Automatic simplification of fractions and roots

Dependencies

To just use the command line interface the only dependency is `lark-parser`, this can be installed with

```
pip install lark-parser
```

Command line usage

To run Caspy you can either execute the `start-caspy.py` file with `python start-caspy.py` or execute the Caspy module with `python -m caspy`. There are some command line arguments that can be passed to Caspy which determines how verbose the output is and the type of the output Caspy will give these are

```
  -h, --help  show this help message and exit
  --timer     Time execution of statements
  --verbose   Enable verbose logging
  --debug     Enable more verbose logging for debugging purposes
  --ascii     Output string using ASCII characters
  --latex     Output representation of string in LaTeX form
  --unicode   Output string using Unicode characters
```

Jupyter kernel

To use the Jupyter kernel interface the `jupyter` module must also be installed, this can be installed with

```
pip install jupyter
```

Then the jupyter kernel can be installed with

`jupyter kernelspec install --user caspy`

Functions implemented

Available functions in the system:

- `integrate(f(x),x)` : Integrate a function with respect to a variable x
- `diff(f(x),x)` : Differentiates a function with respect to a variable x
- `factor(f(x))` : Factorises a polynomial g(x) using Kronecker's algorithm
- `expand_trig(...)` : Expands a trigonometric expression, for instance sin(2x) is expanded as 2sin(x)cos(x)

- `expand(...)` : Expands brackets in an expression
- `re(...)` : Returns floating point answer where possible, for instance re(sqrt(2)) gives 1.4142...

Example usage

Integration

To calculate the integral of (x+e^x)^2 with respect to x in Caspy we just need to use the `integrate(...)` function, as shown bellow

```
>> integrate((x+e^x)^2)
(1/3) · x³ + 2 · e^(x)x - 2 · e^(x) + (1/2) · e^(2 · x)
```

If we wish to integrate with respect to some other variable, say y, we can give a second argument to the `integrate(...)` function, as shown bellow

```
>> integrate(x*sin(y),y)
- cos(y)x
```

Differentiation

The differentiation function `diff(...)` works similarly to the integration function, in that the second, optional, argument specifies what we're differentiating with respect to. For example to differentiate xy^2 with respect to y we exectute the code shown bellow.

```
>> diff(x*y^2,y)
2 · yx
```

Factorisation

The `factor(...)` function computes the irreducible factorisation of a polynomial in x with rational coefficients. For instance to factor x^8-1 we can run the code shown bellow.

```
>> factor(x^8-1)
(-x - 1)(-x + 1)(x² + 1)(x + 1)
```

Conversion to floats

To maintain accuracy and readibility Caspy uses symbolic representations of expressions where possible. This means if we use Caspy to evaluate sin(pi/4), Caspy will return √(2)/2. If we wish to get a floating point representation of this value we can use the `re(...)` function as shown bellow.

```
>> re(sin(pi/4))
0.7071067811865476
```