

The Polycal Package

Tatsuya SATO

2014/01/12 Version 0.01b

Abstract

The `polycal` package adds calculus macros to `polynom` package. It can differentiate or integrate polynomials.

1 Introduction

The `polynom` package manipulates polynomials. It supports multiplication, division, getting GCD of polynomials with Euclidean algorithm, factorization with coefficients in rational numbers. But it lacks in calculus operation, and substitution commands. `Polycal` package adds these commands to `polynom` package.

For one variable polynomial $P(x)$, substitution $P(a)$ can be obtained as a remainder of division with $x - a$. Also dividing with $(x - a)^2$, one can obtain $P'(a)(x - a) + P(a)$ as a remainder, and can get $P'(a)$. But `polynom` package can't deal with multivariate polynomial enough, so sometimes division fails. One can't specify main variable. So I implemented direct substitution commands.

2 Acknowledgments

I wish to thank Carsten Heinz, and Hendri Adriaens for creation and maintainance of `polynom` package.

3 Usage

See the manual of `polynom` package for a basic usage of original commands. This manual explains newly added commands.

3.1 Definition of polynomials

First, specify variables you'll use.

```
\polyset{vars=xab}
```

Define some polynomials using `\polydefine`.

`\polydefine <macro>{<polynomial>}`

`\polydefine\fx{x^2} \polydefine\gx{(a+b)x-ab}`

Polynomials are set to macros `\fx` and `\gx`. You can print them using `\polyprint`:

`\[f(x)=\polyprint\fx,\quad\quad g(x)=\polyprint\gx. \]`

$$f(x) = x^2, \quad g(x) = bx + ax - ab.$$

Never print `\fx` or `\gx` directly.

3.2 Differentiation

You can differentiate polynomials using `\polydiff`.

`\polydiff <macro>{<variable>}{<polynomial>}`

`\polydiff\Dfx{x}\fx`

`\[\frac{df(x)}{dx}=\polyprint\Dfx. \]`

$$\frac{df(x)}{dx} = 2x.$$

If you define `\x` as `\polydefine\x{x}`, you can write this operation as follows.

`\polydiff\aa{x}{x^2} \polydiff\bb{x}\fx`
`\polydiff\cc{x}{x^2} \polydiff\dd{x}\fx`

`\[\polyprint\aa, \polyprint\bb, \polyprint\cc, \polyprint\dd.\]`

$$2x, 2x, 2x, 2x.$$

You may differentiate with another variable:

`\polydiff\Dag{a}\gx \polydiff\Dbg{b}\gx`
`\[\frac{\partial g(x,a,b)}{\partial a} =\polyprint\Dag,\quad`
`\frac{\partial g(x,a,b)}{\partial b} =\polyprint\Dbg. \]`

$$\frac{\partial g(x,a,b)}{\partial a} = x - b, \quad \frac{\partial g(x,a,b)}{\partial b} = x - a.$$

But in this manual, $g(x)$ will be treated as one variable function.

`\polydiff\Dgx{x}\gx \[\frac{dg(x)}{dx}=\polyprint\Dgx. \]`

$$\frac{dg(x)}{dx} = b + a.$$

3.3 Integration

Now we will calculate area between $y = f(x)$ and $y = g(x)$.

```
\polysub\hx\gx\fx
\[ h(x)=g(x)-f(x)=\polyprint\hx. \]
```

$$h(x) = g(x) - f(x) = -x^2 + bx + ax - ab.$$

We will integrate this function.

```
\polyint <macro>\{<variable>\}\{<polynomial>\}
```

This macro calculates indefinite integral of $\langle polynomial \rangle$ from $\langle variable \rangle = 0$.

```
\polyint\Hx{x}\hx \[ H(x)=\int_0^x h(t)dt= \polyprint\Hx. \]
```

$$H(x) = \int_0^x h(t)dt = -\frac{1}{3}x^3 + \frac{1}{2}bx^2 + \frac{1}{2}ax^2 - abx.$$

3.4 Substitution

And definite integral can be obtained by substitution.

```
\polysubst <macro>\{<variable>\}\{<polynomial-1>\}\{<polynomial-2>\}
```

This macro substitute $\langle polynomial-1 \rangle$ for $\langle variable \rangle$ in $\langle polynomial-2 \rangle$.

```
\polysubst\Ha{x}\a\Hx \polysubst\Hb{x}\b\Hx \polysub\S\Hb\Ha
\[ S=\int_a^b h(x)dx=H(b)-H(a)=\polyprint\S. \]
```

$$S = \int_a^b h(x)dx = H(b) - H(a) = \frac{1}{6}b^3 - \frac{1}{6}a^3 + \frac{1}{2}a^2b - \frac{1}{2}ab^2.$$

Unfortunately, factorization of multivariate polynomial is not currently supported, so this can not be arranged to $S = \frac{(b-a)^3}{6}$ automatically.

Let's try another substitution. If you substitute $b = a^2 - 2$ in S , then

```
\polysubst\SS{b}\{a^2-2\}\S
\[ b=a^2-2 \ \ \ \rightarrow \ \ \ S=\polyprint\SS=\polyfactorize\SS. \]
```

$$b = a^2 - 2 \Rightarrow S = \frac{1}{6}a^6 - \frac{1}{2}a^5 - \frac{1}{2}a^4 + \frac{11}{6}a^3 + a^2 - 2a - \frac{4}{3} = \frac{1}{6}(a+1)^3(a-2)^3.$$

```
\polysubstnum <macro>\{<variable>\}\{<rational number>\}\{<polynomial>\}
```

To substitute a number, there's another command. Compared with `\polysubst`, it is based on different algorithm, but not so fast. Experimental.

`\polysubstsqrt` $\langle macro \rangle \{ \langle variable \rangle \} \{ \langle rational number \rangle \} \{ \langle 1\text{-variable polynomial} \rangle \}$

This macro substitute $\sqrt{\langle rational number \rangle}$ for $\langle variable \rangle$. Note that polynomial should contain only one variable. Let substitute $a = \sqrt{5}$ in $S = \frac{1}{6} (a+1)^3 (a-2)^3$.

`\polysubstsqrt\SSS{a}{5}\SS \[S=\polyprint\SSS. \]`

$$S = \left(12 + \frac{-16}{3} \sqrt{5} \right).$$

To eliminate parentheses, set `\polyset{delims={}\{}}`.

`\[\polyset{delims={}\{}}S=\polyprint\SSS. \]`

$$S = 12 + \frac{-16}{3} \sqrt{5}.$$

4 Implementation

Let's start with identification.

```
1 \*package
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{polycal}[2014/01/11 v0.01b (TS)]
4 \RequirePackage{polynom}
```

4.1 The user commands

`\polydefine` $\langle macro \rangle \{ \langle polynomial \rangle \}$

Transform $\langle polynomial \rangle$ to internal representation, and set it to $\langle macro \rangle$.

```
5 \newcommand*\polydefine[2]{\pld@GetPoly#1\{#\2}}
```

`\polydiff` $\langle macro \rangle \{ \langle variable \rangle \} \{ \langle polynomial \rangle \}$

Differentiate $\langle polynomial \rangle$ with respect to $\langle variable \rangle$, and set it to $\langle macro \rangle$.

```
6 \newcommand*\polydiff[1]{%
7   \pld@GetPoly{\pld@polya\pld@polyb}%
8   {\pld@DiffPoly#1\pld@polya\pld@polyb
9     \ignorespaces}}
```

`\polyint` $\langle macro \rangle \{ \langle variable \rangle \} \{ \langle polynomial \rangle \}$

Integrate $\langle polynomial \rangle$ from 0, with respect to $\langle variable \rangle$.

```

10 \newcommand*\polyint[1]{%
11   \pld@GetPoly{\pld@polya\pld@polyb}%
12   {\pld@IntPoly#1\pld@polya\pld@polyb
13     \ignorespaces}}

\polysubstnum <macro>{<variable>}{<rational number>}{<polynomial>}
Substitute <rational number> for <variable> in <polynomial>.

14 \newcommand*\polysubstnum[1]{%
15   \pld@GetPoly{\pld@polya\pld@polyb\pld@polyc}%
16   {\pld@SubstNumPoly#1\pld@polya\pld@polyb\pld@polyc
17     \ignorespaces}}

\polypower <macro>{<variable>}{<polynomial>}{<non negative integer>}
Set <non negative integer>th power of <polynomial> to <macro>.

18 \newcommand*\polypower[1]{%
19   \pld@GetPoly{\pld@polya}%
20   {\pld@PowerPoly#1\pld@polya}}

\polysubst <macro>{<variable>}{<polynomial-1>}{<polynomial-2>}
Substitute <polynomial-1> for <variable> in <polynomial-2>.

21 \newcommand*\polysubst[1]{%
22   \pld@GetPoly{\pld@polya\pld@polyb\pld@polyc}%
23   {\pld@SubstPolyPoly#1\pld@polya\pld@polyb\pld@polyc
24     \ignorespaces}}

\polysubstsqr <macro>{<variable>}{<rational number>}{<1-variable polynomial>}
Substitute non negative square root of <rational number> for <variable> in
<1-variable polynomial>.

25 \newcommand*\polysubstsqr[1]{%
26   \pld@GetPoly{\pld@polya\pld@polyb\pld@polyc}%
27   {\pld@SubstSqrPoly#1\pld@polya\pld@polyb\pld@polyc
28     \ignorespaces}}

```

4.2 Internal routines

The following three macros gets parameter in internal representation of factors.

```

\@fifthofsix For example, \polydefine\X{X} is equivalent to \def\X{\pld@R{1}{1}\pld@V{X}{1}},
\@secondofthree and \@fifthofsix\X gets variable letter X.
\@thirdofthree
29 \providecommand\@fifthofsix[6]{#5}
30 \providecommand\@secondofthree[3]{#2}
31 \providecommand\@thirdofthree[3]{#3}

```

4.2.1 Differentiation

`\pld@DiffPoly` $\langle macro a \rangle \langle macro b \rangle \langle macro c \rangle$

$\langle macro a \rangle$ gets the derivative $\frac{d\langle macro c \rangle}{d\langle macro b \rangle}$.

```

32 \def\pld@DiffPoly#1#2#3{%
33     \let\polycalc@tempb\@empty%
34     \ifx\@empty#3\relax\else
35         \expandafter\pld@DiffPoly@\expandafter#2#3+\relax+%
36     \fi
37     \let#1\polycalc@tempb \pld@Simplify{#1}%
38 }
```

`\pld@DiffPoly@` This macro cuts away a monomial in the third argument $\langle macro c \rangle$ of `\pld@DiffPoly`, send it to `\pld@DiffMonom`.

```

39 \def\pld@DiffPoly@#1#2+{%
40     \let\polycalc@tempa\@empty
41     \ifx\relax#2 \relax
42     \else
43         \pld@DiffMonom\polycalc@tempa#1{#2}%
44         \pld@ExtendPoly\polycalc@tempb{\polycalc@tempa}%
45         \expandafter\pld@DiffPoly@\expandafter#1%
46     \fi}
```

`\pld@DiffMonom` $\langle macro a \rangle \langle macro b \rangle \langle macro c \rangle$

This macro decrease the degree of the monomial $\langle macro c \rangle$ by 1 with respect to variable $\langle macro b \rangle$ and send it to `\pld@DiffMonom@`.

```

47 \def\pld@DiffMonom#1#2#3{%
48     \let#1\@empty
49     \edef\pldc@var@letter{\expandafter\@fifthofsix#2}%
50     \def\@tempb{#3}\edef\@tempc{\noexpand\pld@V{\pldc@var@letter}{-1}}%
51     \pld@MultiplyPoly\@tempa\@tempb\@tempc%
52     \def\pldc@CallMono@##1{%
53         \expandafter\pld@DiffMonom@\expandafter #1\expandafter##1%
54         \@tempa\relax\@empty\@empty\relax\relax}%
55     \expandafter\pldc@CallMono@\pldc@var@letter
56 }
```

`\pld@DiffMonom@` $\langle macro a \rangle \langle variable letter \rangle \langle macro b \rangle \{ \langle content c \rangle \} \{ \langle content d \rangle \}$

This macro cuts away a factor $\langle macro b \rangle \{ \langle content c \rangle \} \{ \langle content d \rangle \}$, of the monomial argument $\langle macro c \rangle$ of `\pld@DiffMonom`. If the factor is a power of $\langle variable letter \rangle$, do $x^{n-1} \rightarrow nx^{n-1}$.

```

57 \def\pld@DiffMonom@#1#2#3#4#5{%
58     \ifx\relax#3
59     \relax \expandafter\@gobbletwo
60     \else
```

```

61      \ifx #3\pld@V \if#4#2
62          \@tempcnta #5 \advance\@tempcnta\@ne
63          \expandafter\pld@Extend\expandafter#1{\expandafter\pld@R\expandafter{\the\@tempcnta}}
64      \fi\fi
65      \pld@Extend#1{\noexpand#3{#4}{#5}}%
66      \expandafter\pld@DiffMonom@\expandafter#1\expandafter#2%
67      \fi}

```

4.2.2 Integration

`\pld@IntPoly` $\langle macro\ a \rangle \langle macro\ b \rangle \langle macro\ c \rangle$

$\langle macro\ a \rangle$ gets the integration $\int_0^{\langle macro\ b \rangle} \langle macro\ c \rangle d\langle macro\ b \rangle$.

```

68 \def\pld@IntPoly#1#2#3{%
69     \let\polycalc@tempb\@empty
70     \ifx\@empty#3\relax\else
71         \expandafter\pld@IntPoly@\expandafter#2#3+\relax+%
72     \fi
73     \let#1\polycalc@tempb \pld@Simplify{#1}%
74 }

```

`\pld@IntPoly@` This macro cuts away a monomial in the third argument $\langle macro\ c \rangle$ of `\pld@IntPoly`, send it to `\pld@IntMonom`.

```

75 \def\pld@IntPoly@#1#2+{%
76     \let\polycalc@tempa\@empty
77     \ifx\relax#2 \relax
78     \else
79         \pld@IntMonom\polycalc@tempa#1{#2}%
80         \pld@ExtendPoly\polycalc@tempb{\polycalc@tempa}%
81         \expandafter\pld@IntPoly@\expandafter#1%
82 \fi}

```

`\pld@IntMonom` $\langle macro\ a \rangle \langle macro\ b \rangle \langle macro\ c \rangle$

This macro increase the degree of the monomial $\langle macro\ c \rangle$ by 1 with respect to variable $\langle macro\ b \rangle$ and send it to `\pld@IntMonom@`.

```

83 \def\pld@IntMonom#1#2#3{%
84     \let#1\@empty
85     \def\@tempb{#3}\def\@tempc{#2}%
86     \pld@MultiplyPoly\@tempa\@tempb\@tempc
87     \edef\pldc@var@letter{\expandafter\@fifthofsix#2}%
88     \def\pldc@CallMono@##1{%
89         \expandafter\pld@IntMonom@\expandafter #1\expandafter##1%
90         \@tempa\relax\@empty\@empty\relax\relax}%
91     \expandafter\pldc@CallMono@\pldc@var@letter
92 }

```

`\pld@IntMonom@` $\langle macro\ a \rangle \langle variable\ letter \rangle \langle macro\ b \rangle \{ \langle content\ c \rangle \} \{ \langle content\ d \rangle \}$

This macro cuts away a factor $\langle macro\ b \rangle \{ \langle content\ c \rangle \} \{ \langle content\ d \rangle \}$, of the monomial argument $\langle macro\ c \rangle$ of `\pld@IntMonom`. If the factor is a power of $\langle variable\ letter \rangle$, do $x^{n+1} \rightarrow \frac{x^{n+1}}{n+1}$.

```

93 \def\pld@IntMonom#1#2#3#4#5{%
94   \ifx\relax#3 \relax
95   \relax \expandafter\@gobbletwo
96   \else
97     \ifx #3\pld@V \if#4#2
98       \pld@Extend#1{\noexpand\pld@R{1}{#5}}%
99       \fi\fi
100     \pld@Extend#1{\noexpand#3{#4}{#5}}%
101     \expandafter\pld@IntMonom@\expandafter#1\expandafter#2%
102   \fi}

```

4.2.3 Substitution of number

`\pld@SubstNumPoly` $\langle macro\ a \rangle \langle macro\ b \rangle \langle macro\ c \rangle \langle macro\ d \rangle$

Substitute rational number $\langle macro\ c \rangle$ for variable $\langle macro\ b \rangle$ in the polynomial $\langle macro\ d \rangle$, and stores it to $\langle macro\ a \rangle$.

```

103 \def\pld@SubstNumPoly#1#2#3#4{%
104   \let\polycalc@tempb\@empty
105   \ifx\@empty#4\relax\else
106     \expandafter\pld@SubstNumPoly@\expandafter#2\expandafter#3#4+\relax+%
107   \fi
108   \let#1\polycalc@tempb \pld@Simplify{#1}%
109 }

```

`\pld@SubstNumPoly@` This macro cuts away a monomial in the fourth argument $\langle macro\ d \rangle$ of `\pld@SubstNumPoly`, send it to `\pld@SubstNumMonom`.

```

110 \def\pld@SubstNumPoly@#1#2#3+{%
111   \let\polycalc@tempa\@empty
112   \ifx\relax#3 \relax
113   \else
114     \pld@SubstNumMonom\polycalc@tempa#1#2{#3}%
115     \pld@ExtendPoly\polycalc@tempb{\polycalc@tempa}%
116     \expandafter\pld@SubstNumPoly@\expandafter#1\expandafter#2%
117   \fi}

```

`\pld@SubstNumMonom` $\langle macro\ a \rangle \langle macro\ b \rangle \langle macro\ c \rangle \langle macro\ d \rangle$

This macro only replace $\langle macro\ b \rangle$ with its variable letter, and sends four arguments to `\pld@SubstNumMonom@`.

```

118 \def\pld@SubstNumMonom#1#2#3#4{%
119   \let#1\@empty
120   \edef\pldc@var@letter{\expandafter\@fifthofsix#2}%
121   \def\pldc@CallMono@##1{%

```



```

122 \expandafter\pld@SubstNumMonom@\expandafter#1\expandafter##1\expandafter#3\expandafter%
123 #4\relax\@empty\@empty\relax\relax}%
124 \expandafter\pldc@CallMono@\pldc@var@letter
125 }

```

`\pld@SubstNumMonom@` $\langle macro\ a \rangle \langle variable\ letter \rangle \langle macro\ b \rangle \langle macro\ c \rangle \{ \langle contents\ d \rangle \} \{ \langle contents\ e \rangle \}$

This macro cuts away a factor $\langle macro\ c \rangle \{ \langle content\ d \rangle \} \{ \langle content\ e \rangle \}$, of the monomial argument $\langle macro\ d \rangle$ of `\pld@SubstNumMonom`. If the factor contains $\langle variable\ letter \rangle$ as variable, this macro substitute the rational number $\langle macro\ b \rangle$ for that variable.

```

126 \def\pld@SubstNumMonom@#1#2#3#4#5#6{%
127   \ifx\relax#4 \relax \expandafter\@gobbletwo
128   \else
129     \ifx #4\pld@V
130       \if#5#2
131         \def\@tempb{\pld@R{-1}{1}}\def\@tempc{#3}%
132         \pld@MultiplyPoly\@tempa\@tempb\@tempc
133         \edef\@tempb{\noexpand\pld@V{#5}{1}}%
134         \pld@ExtendPoly\@tempb\@tempa \pld@Simplify\@tempb
135         \def\@tempc{\pld@V{#5}{#6}}%
136         \let\pld@stage\maxdimen
137         \pld@DividePoly\@tempc\@tempb \pld@Simplify\pld@remainder
138         \pld@Extend#1{\pld@remainder}%
139       \else
140         \pld@Extend#1{\noexpand#4{#5}{#6}}%
141       \fi
142     \else
143       \pld@Extend#1{\noexpand#4{#5}{#6}}%
144     \fi
145   \expandafter\pld@SubstNumMonom@\expandafter#1\expandafter#2\expandafter#3%
146   \fi}

```

4.2.4 Substitution of polynomial

`\pld@SubstPolyPoly` $\langle macro\ a \rangle \langle macro\ b \rangle \langle macro\ c \rangle \langle macro\ d \rangle$

Substitute polynomial $\langle macro\ c \rangle$ for variable $\langle macro\ b \rangle$ in the polynomial $\langle macro\ d \rangle$, and stores it to $\langle macro\ a \rangle$.

```

147 \def\pld@SubstPolyPoly#1#2#3#4{%
148   \let\polycalc@tempb\@empty
149   \ifx\@empty#4\relax\else
150     \expandafter\pld@SubstPolyPoly@\expandafter#2\expandafter#3#4+\relax+%
151   \fi
152   \let#1\polycalc@tempb \pld@Simplify{#1}%
153 }

```

`\pld@SubstPolyPoly@` This macro cuts away a monomial in the fourth argument $\langle macro\ d \rangle$ of `\pld@SubstPolyPoly`, send it to `\pld@SubstPolyMonom`.

```

154 \def\pld@SubstPolyPoly@#1#2#3+{%
155   \let\polycalc@tempa\@empty
156   \ifx\relax#3 \relax
157   \else
158     \pld@SubstPolyMonom\polycalc@tempa#1#2{#3}%
159     \pld@ExtendPoly\polycalc@tempb{\polycalc@tempa}%
160     \expandafter\pld@SubstPolyPoly@\expandafter#1\expandafter#2%
161 \fi}

```

`\pld@SubstPolyMonom` $\langle macro\ a\rangle\langle macro\ b\rangle\langle macro\ c\rangle\langle macro\ d\rangle$

This macro only replace $\langle macro\ b\rangle$ with its variable letter, and sends four arguments to `\pld@SubstPolyMonom@`.

```

162 \def\pld@SubstPolyMonom#1#2#3#4{%
163   \def#1{\pld@R{1}{1}}%
164   \edef\@tempb{\expandafter\@fifthofsix#2}%
165   \edef\pldc@var@letter{\expandafter\@fifthofsix#2}%
166   \def\pldc@CallMono@##1{%
167     \expandafter\pld@SubstPolyMonom@\expandafter#1\expandafter##1\expandafter#3\expandafter%
168     #4\relax\@empty\@empty\relax\relax}%
169   \expandafter\pldc@CallMono@\pldc@var@letter
170 }

```

`\pld@SubstPolyMonom@` $\langle macro\ a\rangle\langle variable\ letter\rangle\langle macro\ b\rangle\langle macro\ c\rangle\{\langle contents\ d\rangle\}\{\langle contents\ e\rangle\}$

This macro cuts away a factor $\langle macro\ c\rangle\{\langle content\ d\rangle\}\{\langle content\ e\rangle\}$, of the monomial argument $\langle macro\ d\rangle$ of `\pld@SubstPolyMonom`. If the factor contains $\langle variable\ letter\rangle$ as variable, this macro substitute the polynomial $\langle macro\ b\rangle$ for that variable.

```

171 \def\pld@SubstPolyMonom@#1#2#3#4#5#6{%
172   \ifx\relax#4 \relax \expandafter\@gobbletwo
173   \else
174     \ifx #4\pld@V
175       \if#5#2
176         \edef\@tempa{\number#6}%
177         \pld@PowerPoly\@tempb#3\@tempa
178         \expandafter\def\expandafter\@tempa\expandafter{#1}%
179         \pld@MultiplyPoly#1\@tempa\@tempb
180       \else
181         \expandafter\def\expandafter\@tempa\expandafter{#1}%
182         \edef\@tempb{\noexpand#4{#5}{#6}}%
183         \pld@MultiplyPoly#1\@tempa\@tempb
184       \fi
185     \else
186       \expandafter\def\expandafter\@tempa\expandafter{#1}%
187       \edef\@tempb{\noexpand#4{#5}{#6}}%
188       \pld@MultiplyPoly#1\@tempa\@tempb
189     \fi
190   \expandafter\pld@SubstPolyMonom@\expandafter#1\expandafter#2\expandafter#3%
191 \fi}

```

`\pld@PowerPoly` $\langle macro a \rangle \langle macro b \rangle \{ \langle number a \rangle \}$

$\langle macro a \rangle$ gets a $\langle number a \rangle$ th power of polynomial $\langle macro b \rangle$. $\langle number a \rangle$ should be a non negative integer.

```

192 \def\pld@PowerPoly#1#2#3{%
193   \ifnum#3=\z@ \def#1{\pld@R{1}{1}}%
194   \else
195     \@tempcnta#3\advance\@tempcnta\m@ne
196     \edef\@tempa{\the\@tempcnta}%
197     \pld@PowerPoly#1#2\@tempa
198     \let\@tempb#1
199     \pld@MultiplyPoly#1\@tempb#2%
200   \fi
201 }
```

4.2.5 Substitution of square root

`\pld@SubstSqrtPoly` $\langle macro a \rangle \langle macro b \rangle \langle macro c \rangle \langle macro d \rangle$

Substitute non negative square root of rational number $\langle macro c \rangle$ for variable $\langle macro b \rangle$ in the **one variable** polynomial $\langle macro d \rangle$, and stores it to $\langle macro a \rangle$.

```

202 \def\pld@SubstSqrtPoly#1#2#3#4{%
203   \edef\pldc@var@letter{\expandafter\@fifthofsix#2}%
204   \edef\pldc@numerator@sqrt{\expandafter\@secondofthree#3}%
205   \edef\pldc@denominator@sqrt{\expandafter\@thirdofthree#3}%
206   \edef\pld@polyd{%
207     \noexpand\pld@R{1}{1}\noexpand\pld@V{\pldc@var@letter }{2}%
208     +\noexpand\pld@R{-\pldc@numerator@sqrt }{\pldc@denominator@sqrt }}%
209   \let\pld@stage\maxdimen
210   \pld@DividePoly#4\pld@polyd
211   \let\pldc@main@formula\pld@remainder
212   \pld@FDefSqrt{\pldc@numerator@sqrt}{\pldc@denominator@sqrt}\let\pldc@sqrt@inner\pld@temp
213   \def\pldc@sqrt@outer{\pld@R {1}{1}\pld@S {\pldc@sqrt@inner }{1}}%
214   \pld@SubstNumPoly#1#2\pldc@sqrt@outer\pldc@main@formula%
215 }
```

4.3 Miscellany

`\pld@ifMonomL@` A bug fix of `\pld@ifMonomL@` in `polynom.sty`.

```

216 %%%% The following is a bug fix of \pld@ifMonomL@ in polynom.sty.
217 \def\pld@ifMonomL@#1\pld@V#2#3#4\@empty#5\pld@V#6#7#8\@empty{%
218   \let\pld@next\@empty
219   \ifx #6\relax \aftergroup\@secondoftwo
220   \else \ifx #2\relax \aftergroup\@firstoftwo
221   \else
222     \def\pld@va{#2}\def\pld@vb{#6}%
223     \ifx\pld@va\pld@vb
224       \ifnum#3=#7\relax
```

```

225         \def\pld@next{\pld@ifMonomL@#4\@empty#8\@empty}%
226     \else
227         \ifnum#3<#7\relax \aftergroup\@firstoftwo
228             \else \aftergroup\@secondoftwo \fi
229     \fi
230 \else
231     % \pld@ifVarL#2\relax\@empty#6\relax\@empty % Bug code.
232     \pld@ifVarL#2#6% Fixed code.
233     {\aftergroup\@firstoftwo}%
234     {\aftergroup\@secondoftwo}%
235 \fi
236 \fi \fi
237 \pld@next}
238 \end{package}

```

5 History

0.01b 2014/01/12

- Wrote .dtx.
- Minor fix in \pld@SubstSqrtPoly.

0.01a 2014/01/11

- Fixed a bug in \pld@ifMonomL@ of original polynom package.

0.01 2014/01/09 (private test version)

- Implemented \pld@DiffPoly, \pld@IntPoly, \pld@SubstNumPoly, \pld@SubstPlyPoly, \pld@SubstSqrtPoly.

TODO:

- User definable order of variables.
- Rational functions and its derivatives.
- Vector calculus.
- Differential geometric operations.