1 Polygons algebraic integers

We develop code to operate complicated numbers appearing in polygons constructions, like the number:

$$\cos\frac{2\pi}{17} = \frac{-1+\sqrt{17}+\sqrt{34-2\sqrt{17}}+2\sqrt{17+3\sqrt{17}-\sqrt{170+38\sqrt{17}}}}{16} \tag{1.1}$$

Define A_0, A_1, A_2 and A_3 algebraic integers in the numerator of equation (1.1):

$$A_0 = \pm b \tag{1.2}$$
$$A_1 = \pm c\sqrt{\pm d} \tag{1.3}$$
$$A_2 = \pm e\sqrt{f \pm g\sqrt{\pm h}} \tag{1.4}$$
$$A_3 = \pm i\sqrt{j \pm k\sqrt{\pm l} \pm m\sqrt{\pm n \pm o\sqrt{\pm p}}} \tag{1.5}$$

Define a function F and apply it to the variables above b to p:

$$F = F(x, F, F, ..., F) \tag{1.6}$$
$$= x\sqrt{F + F + ... + F} \tag{1.7}$$
$$F_0(b) = F(b) \tag{1.8}$$
$$F_1(c, d) = F(c, F_0(d)) \tag{1.9}$$
$$F_2(e, f, g, h) = F(e, F_0(f), F_1(g, h)) \tag{1.10}$$
$$F_3(i, j, k, l, m, n, o, p) = F(i, F_0(j), F_1(k, l), F_2(m, n, o, p)) \tag{1.11}$$

So first equation can be expressed as:

$$\cos\frac{2\pi}{17} = \frac{F_0(-1) + F_1(1, 17) + F_2(1, 34, -2, 17) + F_3(2, 17, 3, 17, -1, 170, 38, 17)}{16} \tag{1.12}$$

1.1 plan

Simplification:

$$F_3(1, 1, 1, 2, 1, 2, 1, 1) = 1\sqrt{1 + 1\sqrt{2} + 1\sqrt{2 + 1\sqrt{1}}} \tag{1.13}$$
$$= \sqrt{1 + \sqrt{2} + \sqrt{3}} \tag{1.14}$$

1.2 32 bits limits

Algebraic integers **parts** are expresed as $AZ32$ golang objects. The variable o (see line 2) is the number's part outside square root. The array i (see line 3) is the number's parts sum inside square root. Symbollicaly:

$$a = F(o, F(o, ...), ...) \tag{1.15}$$
$$n = len(a.i) \tag{1.16}$$
$$value = \begin{cases} a.o & n = 0 \\ a.o\sqrt{\sum_{j=0}^{n} a.i[j]} & n > 0 \end{cases} \tag{1.17}$$

```
1 type AZ32 struct {
2   o int32
3   i []*AZ32
4 }
5
6 type AZ32s struct { // factory
7
8 }
```

```

func (a *AZ32s) F0(b int32) *AZ32 {
  return &AZ32{
    o: b,
  }
}

func (a *AZ32s) F1(c, d int32) *AZ32 {
  return &AZ32 {
    o: c,
    i: []*AZ32{
      a.F0(d),
    },
  }
}

func (a *AZ32s) F2(e, f, g, h int32) *AZ32 {
  return &AZ32 {
    o: e,
    i: []*AZ32{
      a.F0(f),
      a.F1(g, h),
    },
  }
}

func (a *AZ32s) F3(i, j, k, l, m, n, o, p int32) *AZ32 {
  return &AZ32 {
    o: i,
    i: []*AZ32{
      a.F0(j),
      a.F1(k, l),
      a.F2(m, n, o, p),
    },
  }
}
```

1.3 N32, I32, AI32, AQ32

```
1  type N32 uint32 // range 0 - 0xffffffff
2
3  type I32 struct {
4    s bool // sign: true means negative
5    n N32  // positive value
6  }
7
8  type AI32 struct {
9    o *I32  // outside radical
10   i *I32  // inside radical square-free
11   e *AI32 // inside radical extension
12 }
13
14 type AQ32 struct {
15   nums []*AI32 // numerator sum
16   den  N32     // denominator
17 }
```

In this list we define four 32 bit numbers in Golang code.
In line 1 we define the natural number $N32$ with a range of $0 < n \leq 2^{32} - 1$.
In line 3 we define the integer number $I32$, the number sign is negative if s is true and the number value always is a positive. If $I32$ is nil, then we assume the number is zero.
In line 8 we define the algebraic integer number $AI32$. The number is recursive with a value of

$$\pm o\sqrt{\pm i \pm e.o\sqrt{\pm e.i \pm e.e.o...}} \tag{1.18}$$

where each sign $\pm$ corresponds to its integer sign s of the values of integers o and i; a $AI32$ with value nil corresponds to a zero. In line 14 we define the algebraic rational number $AQ32$. In the numerator has a sum of several integer numbers $AI32$ and in the denominator a natural number $N32$ other than zero.

1.4 Reductions $R32$

Reductions factory $R32$ produce irreducibles $AI32$ and $AQ32$ using a precomputed fixed list of 32-bit primes. The reduction simplify and standarize the algebraic numbers for further operations like cloning, addition, multiplication, inversion and square root extractions with results also reduced.

```
type Red32 struct {
  primes []N32
}

func NewRed32() *Red32 {
  value := 0xffff
    f := make([]bool, value)
    for i := 2; i <= int(math.Sqrt(float64(value))); i++ {
        if f[i] == false {
            for j := i * i; j < value; j += i {
                f[j] = true
            }
        }
    }
    primes := make([]N32, 0)
    for i := N32(2); i < N32(value); i++ {
        if f[i] == false {
            primes = append(primes, i)
        }
    }
  return &Red32{
    primes: primes,
  }
}
```

Numbers of the form $x\sqrt{y}$ are reduced with a function call named $roi(x, y)$. Numbers of the form $x\sqrt{y + z\sqrt{...}}$ are reduced with function named $roie(x, y, z)$. The factory produces $AI32$ numbers calling both functions as necessary to return irreducible algebraic integers. As an example, this is the process to reduce number A_3:

$$A_3 = \pm i\sqrt{\pm j \pm k\sqrt{\pm l \pm m\sqrt{\pm n}}} \tag{1.19}$$

$$\boxed{m_1, n_1 = roi(m, n)} \tag{1.20}$$

$$= \pm i\sqrt{\pm j \pm k\sqrt{\pm l \pm m_1\sqrt{\pm n_1}}} \tag{1.21}$$

$$\boxed{k_1, l_1, m_2 = roie(k, l, m_1)} \tag{1.22}$$

$$= \pm i\sqrt{\pm j \pm k_1\sqrt{\pm l_1 \pm m_2\sqrt{\pm n_1}}} \tag{1.23}$$

$$\boxed{i_1, j_1, k_2 = roie(i, j, k_1)} \tag{1.24}$$

$$= \pm i_2\sqrt{\pm j_2 \pm k_3\sqrt{\pm l_1 \pm m_2\sqrt{\pm n_1}}} \tag{1.25}$$

1.5 Reduction roi

This reduction is done for $AI32$ numbers parts **without** extension e. This is the case of whole part $\pm c\sqrt{\pm d}$ of A_1, part $\pm g\sqrt{\pm h}$ of A_2 and part $\pm m\sqrt{\pm n}$ of A_3. Example of reducing $A_1 = \pm c\sqrt{\pm d}$. From d find two numbers p and

d_1, so p is the product of some primes even repeated and d_1 is square-free or 1:

$$A_1 = \pm c\sqrt{\pm d} \tag{1.26}$$

$$\boxed{d = p^2 d_1} \tag{1.27}$$

$$A_1 = \begin{cases} 0 & \text{case 1: if } c = 0 \text{ or } d = 0 \\ \pm cp\sqrt{+1} & \text{case 2: if } d_1 = +1 \\ \pm c\sqrt{\pm d} & \text{case 3: if } p = 1 \\ \pm cp\sqrt{\pm d/p} & \text{case 4: otherwise} \end{cases} \tag{1.28}$$

For case 1 and case 2 we got A_1 degenerated into A_0. For case 3 the number remains the same because was irreducible. For case 4 the reduction updates the values where $p > 1$ and $c_1 = cp$ and $d_1 = d/p$.

```
func (r *Red32) roi(out, inA Z) (ai *AI32, overflow bool) {
  if out == 0 || inA == 0 { // case 1
    return nil, false // zero
  }
  io := out; if out < 0 { io = -out }
  ia := inA; if inA < 0 { ia = -inA }
  if no, na, overflow := r.roiN(N(io), N(ia)); overflow {
    return nil, true
  } else { // cases 2,3,4
    return &AI32{
      o: &I32{ n:no, s:out < 0 },
      i: &I32{ n:na, s:inA < 0 },
    }, false
  }
}

func (r *Red32) roiN(out, in N) (o N32, i N32, overflow bool) {
  if out == 0 || in == 0 {
    return 0, 0, false // zero
  }
  for _, prime := range r.primes {
    p := N(prime)
    if pp := p*p; in >= pp {
      for {
        if in % pp == 0 { // reduce ok
          out *= p
          in  /= pp
          continue // look for repeated squares in reduced in
        }
        break // check next prime
      }
    } else { // in has no more factors to check
      break
    }
  }
  if out > N32_MAX || in > N32_MAX {
    return 0, 0, true // overflow
  }
  return N32(out), N32(in), false
}
```

1.6 Reduction *roie*

This reduction is done for $AI32$ numbers **with** extension e. This is the case of part $\pm e\sqrt{\pm f + g\sqrt{\ldots}}$ of A_2, part $\pm i\sqrt{\pm j + k\sqrt{\ldots}}$ of A_3 and part $\pm k\sqrt{\pm l + m\sqrt{\ldots}}$ of A_3. Example of reducing A_2. First we reduce part $\pm g\sqrt{\pm h}$ with

$roi(g, h)$ and apply the four cases to A2:

$$A_1 = \pm g\sqrt{\pm h} \tag{1.29}$$

$$\boxed{h = p^2 h_1} \tag{1.30}$$

$$A_2 = \pm e\sqrt{f \pm gp\sqrt{\pm h_1}} \tag{1.31}$$

$$A_2 = \begin{cases} 0 & case5 : \text{if } e = 0 \\ \pm e\sqrt{\pm f} & case6 : \text{if } g = 0 \text{ or } h = 0 \\ \pm e\sqrt{\pm f \pm gp} & case7 : \text{if } h_1 = +1 \\ \pm e\sqrt{\pm f \pm gp\sqrt{\pm h_1}} & case8 : \text{if } p \geq 1 \end{cases} \tag{1.32}$$

For case 5 we have that A_2 is zero. For case 6 we reduce A_2 with $roi(\pm e, \pm f)$. For case 7 we reduce A_2 with $roi(\pm e, \pm f \pm gp)$.

For case 8 we reduce A_2 with $roia(\pm e, \pm f, \pm gp)$ where h_1 is irreducible. Reduction $roie$ look for another product of primes q such that at the same time $f = q^2 f_1$ and $gp = q^2 g_1$:

$$A_2 = \pm e\sqrt{\pm f \pm gp\sqrt{\pm h1}} \tag{1.33}$$

$$\boxed{f = q^2 f_1} \tag{1.34}$$

$$\boxed{gp = q^2 g_1} \tag{1.35}$$

$$A_2 = \pm e_1\sqrt{\pm f_1 \pm g_1\sqrt{\pm h_1}} \tag{1.36}$$

Where $e_1 = eq$, $f_1 = f/q^2$, $g_1 = gp/q^2$.

```
func (r *Red32) roie(out, inA, inB Z) (o, i, j Z, overflow bool) {
  if out == 0 { // case 1
    return 0, 0, 0, false // zero
  }
  io := out; if out < 0 { io = -out }
  ia := inA; if inA < 0 { ia = -inA }
  ib := inB; if inB < 0 { ib = -inB }
  if no, na, nb, overflow := r.roieN(N(io), N(ia), N(ib)); overflow {
    return 0, 0, 0, true // overflow
  } else {
    zo := Z(no); if out < 0 { zo = -zo }
    za := Z(na); if inA < 0 { za = -za }
    zb := Z(nb); if inB < 0 { zb = -zb }
    return zo, za, zb, false
  }
}

func (r *Red32) roieN(out, inA, inB N) (o, i, j N32, overflow bool) {
  if inA > 1 && inB > 1 {
    for _, prime := range r.primes {
      p := N(prime)
      pp := p*p
      for {
        if inA % pp == 0 && inB % pp == 0 {
          out *= p  // multiply by p
          inA /= pp // divide by p squared
          inB /= pp // divide by p squared
          continue
        } else {
          break
        }
      }
    }
```

```
  }
  if out > N32_MAX || inA > N32_MAX || inB > N32_MAX {
    return 0, 0, 0, true // overflow
  }
  return N32(out), N32(inA), N32(inB), false
}
```

1.7 B, D, H, N

We define four numbers of increasing complexity:

$$B \equiv \frac{A_0}{a} \tag{1.37}$$

$$D \equiv \frac{A_0 + A_1}{a} \tag{1.38}$$

$$H \equiv \frac{A_0 + A_1 + A_2}{a} \tag{1.39}$$

$$N \equiv \frac{A_0 + A_1 + A_2 + A_3}{a} \tag{1.40}$$

2 functions

Each of the radicals $r_0, ..., r_3$ has a function to read their corresponding signs and integers variables:

$$f_0 \equiv f(\pm b) \tag{2.1}$$

$$f_1 \equiv f(\pm c, d) \tag{2.2}$$

$$f_2 \equiv f(\pm e, f, \pm g, h) \tag{2.3}$$

$$f_3 \equiv f(\pm i, j, \pm k, l, \pm m, n) \tag{2.4}$$

Each $f_0, ...f_4$ reduces the values with gcd and root simplifications.

Each of the algebraic numbers B, D, H and N has a function to read their radicals functions as inputs:

$$f_B \equiv f(f_0(...), a) \tag{2.5}$$

$$f_D \equiv f(f_0(...), f_1(...), a) \tag{2.6}$$

$$f_H \equiv f(f_0(...), f_1(...), f_2(...), a) \tag{2.7}$$

$$f_N \equiv f(f_0(...), f_1(...), f_2(...), f_3(...), a) \tag{2.8}$$

Each $f_B, ...f_N$ adds the radicals reducing once more the variables with gcd root simplifications and now considering the denominator a.

3 Examples

3.1 f_B examples

$$\cos 0 = 1 \implies f_B(f_0(1), 1) \tag{3.1}$$

$$\sin \frac{\pi}{6} = \frac{1}{2} \implies f_B(f_0(1), 2) \tag{3.2}$$

3.2 f_D examples

$$\sin \frac{\pi}{4} = \frac{\sqrt{2}}{2} \implies f_D(\emptyset, f_1(1, 2), 2) \tag{3.3}$$

$$\sin \frac{\pi}{10} = \frac{-1 + \sqrt{5}}{4} \implies f_D(f_0(-1), f_1(1, 5), 4) \tag{3.4}$$

3.3 f_H examples

$$\sin\frac{\pi}{5} = \frac{\sqrt{10-2\sqrt{5}}}{4} \implies f_H(\emptyset, \emptyset, f_2(1, 10, -2, 5), 4) \tag{3.5}$$

$$\sin\frac{\pi}{12} = \frac{\sqrt{6}+\sqrt{2}}{4} \implies f_H(\emptyset, f_1(1, 6), f_2(1, 2, 0, 0), 4)* \tag{3.6}$$

$$\sin\frac{\pi}{12} = \frac{\sqrt{2+\sqrt{3}}}{2} \implies f_H(\emptyset, \emptyset, f_2(1, 2, 1, 3), 2) \tag{3.7}$$

$$\cos\frac{\pi}{15} = \frac{1+\sqrt{5}+\sqrt{30-6\sqrt{5}}}{8} \implies f_E(f_0(1), f_1(1, 5), f_2(1, 30, -6, 5), 8) \tag{3.8}$$

3.4 f_N examples

$$\cos\frac{\pi}{16} = \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \implies f_N(\emptyset, \emptyset, \emptyset, f_3(1, 2, 1, 2, 1, 2), 2) \tag{3.9}$$

$$\cos\frac{\pi}{24} = \frac{\sqrt{2+\sqrt{2+\sqrt{3}}}}{2} \implies f_N(\emptyset, \emptyset, \emptyset, f_3(1, 2, 1, 2, 1, 3), 2) \tag{3.10}$$

$$\cos\frac{2\pi}{17} = \frac{-1+\sqrt{17}+\sqrt{34-2\sqrt{17}}+2\sqrt{17+3\sqrt{17}-\sqrt{170+38\sqrt{17}}}}{16} \tag{3.11}$$

$$\implies f_N(f_0(-1), f_1(1, 17), f_2(1, 34, -2, 17), f_3(2, 17, 3, 17, -1, 170, +38, 17), 16) \tag{3.12}$$

4 Operations with result B

4.1 NewB $B = B_1$

$$B_1 = \frac{\pm b_1}{a_1} \tag{4.1}$$

$$\textbf{Reduce } \{a, b\} = \{a_1/G, b_1/G\} \iff G = \gcd\{a_1, b_1\} > 1$$

$$= \frac{\pm b}{a} \tag{4.2}$$

4.2 AddBB $B = B_2 + B_3$

$$B_2 + B_3 = \frac{\pm b_2}{a_2} + \frac{\pm b_3}{a_3} \tag{4.3}$$

$$= \frac{\pm b_2 a_3 \pm b_3 a_2}{a_2 a_3} = \frac{q}{p} \tag{4.4}$$

$$\textbf{Reduce } \{a_1, b_1\} = \{p/G, q/G\} \iff G = \gcd\{p, q\} > 1$$

$$= \frac{\pm b_1}{a_1} \textbf{ Solve as NewB} \tag{4.5}$$

4.3 MulBB $B = B_2 \times B_3$

$$
\begin{aligned}
B_2 \times B_3 &= \frac{\pm b_2}{a_2} \times \frac{\pm b_3}{a_3} && (4.6)\\
&= \frac{\pm b_2 b_3}{a_2 a_3} = \frac{q}{p} && (4.7)\\
&\quad \textbf{Reduce } \{a_1, b_1\} = \{p/G, q/G\} \iff G = \gcd\{p, q\} > 1 \\
&= \frac{\pm b_1}{a_1} \textbf{ Solve as NewB} && (4.8)
\end{aligned}
$$

4.4 InvB $B = 1/B_2$

$$
\begin{aligned}
\frac{1}{B_2} &= \frac{1}{\pm b_2/a_2} && (4.9)\\
&= \frac{\pm a_2}{b_2} = \frac{q}{p} && (4.10)\\
&\quad \textbf{Reduce } \{a_1, b_1\} = \{p/G, q/G\} \iff G = \gcd\{p, q\} > 1 \\
&= \frac{\pm b_1}{a_1} \textbf{ Solve as NewB} && (4.11)
\end{aligned}
$$

5 Operations with result D

5.1 NewD $D = D_1$

$$
\begin{aligned}
D_1 &= \frac{\pm b_1 \pm c_1\sqrt{d_1}}{a_1} && (5.1)\\
&\quad \textbf{Reduce } \{p, q, r\} = \{a_1/G, b_1/G, c_1/G\} \iff G = \gcd\{a_1, b_1, c_1\} > 1 \\
&= \frac{\pm q \pm r\sqrt{d_1}}{p} && (5.2)\\
&\quad \textbf{Reduce } \{d\} = s^2 d_1 \iff s > 1 \\
&= \frac{\pm q \pm rs\sqrt{d}}{p} && (5.3)\\
&\quad \textbf{Reduce } \{a, b, c\} = \{p/G, q/G, rs/G\} \iff G = \gcd\{p, q, rs\} \\
&= \frac{\pm b \pm c\sqrt{d}}{a} && (5.4)
\end{aligned}
$$

5.2 SqrtB $D = \sqrt{B_2}$

$$
\begin{aligned}
\sqrt{B_2} &= \sqrt{\frac{\pm b_2}{a_2}} && (5.5)\\
&= \frac{\sqrt{a_2 b_2}}{a_2} && (5.6)\\
&\quad \textbf{Set } \{a_1, b_1, c_1, d_1\} = \{a_2, 0, 1, a_2 b_2\} \\
&= \frac{\pm b_1 \pm c_1\sqrt{d_1}}{a_1} \textbf{ Solve as NewD} && (5.7)
\end{aligned}
$$

5.3 InvD $D = 1/D_2$

$$\begin{aligned}
1/D_2 &= \frac{a_2}{\pm b_2 \pm c_2\sqrt{d_2}} \\
&= \frac{\pm a_2 b_2 \mp a_2 c_2\sqrt{d_2}}{b_2^2 - c_2^2 d_2} \\
&\quad \textbf{Set } \{a_1, b_1, c_1, d_1\} = \{b_2^2 - c_2^2 d_2, \pm a_2 b_2, \mp a_2 c_2, d_2\} \\
&= \frac{\pm b_1 \pm c_1\sqrt{d_1}}{a_1} \textbf{ Solve as NewD}
\end{aligned}$$

6 Operations with result H

6.1 $D_1 + D_2 \mapsto H$ ¡¡¡¡¡

$$D_1 + D_2 = \frac{\pm b_1 \pm c_1\sqrt{d_1}}{a_1} + \frac{\pm b_2 \pm c_2\sqrt{d_2}}{a_2} \tag{6.1}$$

$$= \frac{(\pm a_2 b_1 \pm a_1 b_2) \pm a_2 c_1\sqrt{d_1} \pm a_1 c_2\sqrt{d_2}}{a_1 a_2} \tag{6.2}$$

$$= \frac{\pm q \pm r\sqrt{d_1} \pm s\sqrt{d_2}}{p} \tag{6.3}$$

where $\{p, q, r, s\} = \gcd\{a_1 a_2, (\pm a_2 b_1 \pm a_1 b_2), \pm a_2 c_1, \pm a_1 c_2\}$

$$= \frac{\pm q \pm \sqrt{r^2 d_1 + s^2 d_2 \pm 2rs\sqrt{d_1 d_2}}}{p} \tag{6.4}$$

$$= \frac{\pm q \pm \sqrt{t \pm 2rsu\sqrt{h}}}{p} \tag{6.5}$$

where $\{t\} = r^2 d_1 + s^2 d_2$ **and** $\{u^2 h\} = d_1 d_2$

$$= \frac{\pm q \pm v\sqrt{f \pm g\sqrt{h}}}{p} \tag{6.6}$$

where $\{v^2 f\} = t$ **and** $\{v^2 g\} = 2rsu$

$$= \frac{\pm d \pm e\sqrt{f \pm g\sqrt{h}}}{a} \tag{6.7}$$

where $\{a, d, e\} = \gcd\{p, \pm q, \pm qv\}$

$$\tag{6.8}$$

6.2 $\sqrt{C_1} = F_2$

$$\begin{aligned}
\sqrt{C_1} &= \sqrt{\frac{a_1\sqrt{c_1}}{b_1}} \\
&= \frac{\sqrt{a_1 b_1\sqrt{c_1}}}{b_1} \\
&= \frac{m\sqrt{e_2\sqrt{c_1}}}{b_1} && a_1 b_1 = m^2 e_2 \\
&= \frac{a_2\sqrt{e_2\sqrt{c_1}}}{b_2} && (a_2, b_2) = gcd(m, b_1)
\end{aligned}$$

6.3 $C_1 + D_2 = F_3$

$$
\begin{aligned}
C_1 + D_2 &= \frac{\pm a_1\sqrt{c_1}}{b_1} + \frac{\pm a_2\sqrt{c_2} \pm d_2}{b_2} \\
&= \frac{\pm a_1 b_2\sqrt{c_1} \pm a_2 b_1\sqrt{c_2} \pm d_2 b_1}{b_1 b_2} \\
&= \frac{\pm m\sqrt{c_1} \pm n\sqrt{c_2} \pm p}{o} && (\pm m, \pm n, \pm p, o) = gcd(\pm a_1 b_2, \pm a_2 b_1, \pm d_2 b_1, b_1 b_2) \\
&= \frac{\sqrt{m^2 c_1 + n^2 c_2 \pm 2mn\sqrt{c1 c_2}} \pm p}{o} \\
&= \frac{\sqrt{q \pm 2mnr\sqrt{f_3}} \pm p}{o} && q = m^2 c_1 + n^2 c_2, c_1 c_2 = r^2 f_3 \\
&= \frac{s\sqrt{c_3 \pm e_3\sqrt{f_3}} \pm p}{o} && q = s^2 c_3, 2mnr = s^2 e_3 \\
&= \frac{a_3\sqrt{c_3 \pm e_3\sqrt{f_3}} \pm d_3}{b_3} && (a_3, b_3, \pm d_3) = gcd(s, \pm p, o)
\end{aligned}
$$

6.4 $1/D_1 = D_2$

$$
\begin{aligned}
1/D_1 &= \frac{b_1}{\pm a_1\sqrt{c_1} \pm d_1} \\
&= \frac{\pm a_1 b_1\sqrt{c_1} \mp b_1 d_1}{a_1^2 c_1 - d_1^2} \\
&= \frac{a_2\sqrt{c_1} \pm d_2}{b_2} && (a_2, b_2, d_2) = gcd(\pm a_1 b_1, \mp b_1 d_1, a_1^2 c_1 - d_1^2)
\end{aligned}
$$

6.5 $\sqrt{D_1} = F_2$ editing...

$$
\begin{aligned}
\sqrt{D_1} &= \sqrt{\frac{\pm a_1\sqrt{c_1} \pm d_1}{b_1}} \\
&= \frac{\sqrt{\pm b_1 d_1 \pm a_1 b_1\sqrt{f_2}}}{b_1} && f_2 = c_1 \\
&= \frac{m\sqrt{c_2 \pm e_2\sqrt{f_2}}}{b_1} && \pm b_1 d_1 = m^2 c_2, \pm a_1 b_1 = m^2 e_2 \\
&= \frac{a_2\sqrt{c_2 \pm e_2\sqrt{f_2}}}{b_2} && (a_2, b_2) = gcd(m, b_1)
\end{aligned}
$$

6.6 $D_1 + D_2 = F_3$

$$\begin{aligned}
D_1 + D_2 &= \frac{\pm a_1\sqrt{c_1} \pm d_1}{b_1} + \frac{\pm a_2\sqrt{c_2} \pm d_2}{b_2} \\
&= \frac{\pm a_1 b_2\sqrt{c_1} \pm a_2 b_1\sqrt{c_2} \pm d_1 b_2 \pm d_2 b_1}{b_1 b_2} \\
&= \frac{\pm m\sqrt{c_1} \pm n\sqrt{c_2} \pm p}{o} && (\pm m, \pm n, \pm p, o) = gcd(\pm a_1 b_2, \pm a_2 b_1, \pm d_1 b_2 \pm d_2 b_1, b_1 b_2) \\
&= \frac{\sqrt{m^2 c_1 + n^2 c_2 \pm 2mn\sqrt{c1 c_2}} \pm p}{o} \\
&= \frac{\sqrt{q \pm 2mnr\sqrt{f_3}} \pm p}{o} && q = m^2 c_1 + n^2 c_2, c_1 c_2 = r^2 f_3 \\
&= \frac{s\sqrt{c_3 \pm e_3\sqrt{f_3}} \pm p}{o} && q = s^2 c_3, 2mnr = s^2 e_3 \\
&= \frac{a_3\sqrt{c_3 \pm e_3\sqrt{f_3}} \pm d_3}{b_3} && (a_3, b_3, \pm d_3) = gcd(s, \pm p, o)
\end{aligned}$$

6.7 $D_1 \times D_2 = F_3$

$$\begin{aligned}
D_1 \times D_2 &= \frac{\pm a_1\sqrt{c_1} \pm d_1}{b_1} \times \frac{\pm a_2\sqrt{c_2} \pm d_2}{b_2} \\
&= \frac{\pm a_1 a_2\sqrt{c_1 c_2} \pm a_1 d_2\sqrt{c1} \pm a_2 d_1\sqrt{c_2} \pm d_1 d_2}{b_1 b_2}
\end{aligned}$$

6.8 MulDD $D_1 \times D_2 \mapsto H$???

$$\begin{aligned}
D_1 \times D_2 &= \frac{\pm b_1 \pm c_1\sqrt{d_1}}{a_1} \times \frac{\pm b_2 \pm c_2\sqrt{d_2}}{a_2} \\
&= \frac{\pm b_1 b_2 \pm b_1 c_2\sqrt{d_2} \pm b_2 c_1\sqrt{d_1} \pm c_1 c_2\sqrt{d_1 d_2}}{a_1 a_2} \\
&= \frac{\pm a_1 a_2 m\sqrt{c_3}}{b_1 b_2} && c_1 c_2 = m^2 c_3 \\
&= \frac{\pm a_3\sqrt{c_3}}{b_3} && (\pm a_3, b_3) = gcd(\pm a_1 a_2 m, b_1 b_2)
\end{aligned}$$