Algorithm 1 F5

globals r, *Rule*, $<_T$
inputs
  $F = (f_1, f_2, \ldots, f_m) \in \mathcal{R}^m$ (homogeneous)
  $<$, an admissible ordering
outputs
  a Gröbner basis of F with respect to $<$
do
  $<_T := <$
  Sort F by increasing total degree, breaking ties by increasing leading monomial
  — Initialize the record keeping.
  $\mathit{Rule} := \mathrm{List}\,(\mathrm{List}\,())$
  $r := \mathrm{List}\,()$
  Append $\left(\mathbf{F}_1, f_1 \cdot \mathrm{lc}\,(f_1)^{-1}\right)$ to r
  — Compute the basis of $\langle f_1 \rangle$.
  $G_{\mathrm{prev}} = \{1\}$
  $B = \{f_1\}$
  — Compute the bases of $\langle f_1, f_2 \rangle, \ldots, \langle f_1, f_2, \ldots, f_m \rangle$.
  $i := 2$
  while $i \leq m$
    Append $\left(\mathbf{F}_i, f_i \cdot \mathrm{lc}\,(f_i)^{-1}\right)$ to r
    $G_{\mathrm{curr}} :=$ INCREMENTAL_BASIS (F5) $(i, B, G_{\mathrm{prev}})$
    if $\exists \lambda \in G_{\mathrm{curr}}$ such that $\mathrm{Poly}\,(\lambda) = 1$
      return $\{1\}$
    $G_{\mathrm{prev}} := G_{\mathrm{curr}}$
    $B := \{\mathrm{Poly}\,(\lambda) : \lambda \in G_{\mathrm{prev}}\}$
    $i := i + 1$
  return B

Algorithm 2 INCREMENTAL_BASIS (F5)

globals r, $<_T$
inputs
  $i \in \mathbb{N}$
  B, a Gröbner basis of $(f_1, f_2, \ldots, f_{i-1})$ with respect to $<_T$
  $G_{\text{prev}} \subset \mathbb{N}$, indices in r of B
outputs
  G_{curr}, indices in r of a Gröbner basis of $(f_1, f_2, \ldots, f_i)$ with respect to $<_T$
do
  $curr\_idx := \#r$
  $G_{\text{curr}} := G_{\text{prev}} \cup \{curr\_idx\}$
  Append List () to *Rule*
  $P := \bigcup_{j \in G_{\text{prev}}} \text{CRITICAL\_PAIR}\,(curr\_idx, j, i, G_{\text{prev}})$
  while $P \neq \emptyset$
    $d := \min\{\deg t : (t, k, u, \ell, v) \in P\}$ — See Algorithm 3 for structure of $p \in P$
    $P_d := \{(t, k, u, \ell, v) \in P : d = \deg t\}$
    $P := P \backslash P_d$
    $S := \text{COMPUTE\_SPOLS}\,(P_d)$
    $R := \text{REDUCTION}\,(S, B, G_{\text{prev}}, G_{\text{curr}})$
    for $k \in R$
      $P := P \cup \left(\bigcup_{j \in G_{\text{curr}}} \text{CRITICAL\_PAIR}\,(k, j, i, G_{\text{prev}})\right)$
      $G_{\text{curr}} := G_{\text{curr}} \cup \{k\}$
  return G_{curr}

Algorithm 3 CRITICAL_PAIR

globals $<_T$
inputs
  $k, \ell \in \mathbb{N}$ such that $1 \leq k < \ell \leq \#r$
  $i \in \mathbb{N}$
  $G_{\text{prev}} \subset \mathbb{N}$, indices in r of a Gröbner basis of $(f_1, f_2, \ldots, f_{i-1})$ w/respect to $<_T$
outputs
  $\{(t, u, k, v, \ell)\}$, corresponding to a critical pair $\{k, l\}$ necessary for
    the computation of a Gröbner basis of $(f_1, f_2, \ldots, f_i)$; $\emptyset$ otherwise
do
  $t_k := \text{lt}\,(\text{Poly}\,(k))$
  $t_\ell := \text{lt}\,(\text{Poly}\,(\ell))$
  $t := \text{lcm}\,(t_k, t_\ell)$
  $u_1 := t/t_k$
  $u_2 := t/t_\ell$
  $\mu_1 \mathbf{F}_{\nu_1} := \text{Sig}\,(k)$
  $\mu_2 \mathbf{F}_{\nu_2} := \text{Sig}\,(\ell)$
  if $\nu_1 = i$ **and** $u_1 \cdot \mu_1$ is top-reducible by G_{prev} — Stegers checks by G_{ν_1+1}
    return $\emptyset$
  if $\nu_2 = i$ **and** $u_2 \cdot \mu_2$ is top-reducible by G_{prev} — Stegers checks by G_{ν_2+1}
    return $\emptyset$
  — A minor optimization is to check IS_REWRITABLE here
  if $u_1 \cdot \text{Sig}\,(k) \prec u_2 \cdot \text{Sig}\,(\ell)$ — Faugère's writeup compares $\text{Sig}(k) \prec \text{Sig}(\ell)$.
    Swap u_1 and u_2
    Swap k and ℓ
  return $\{(t, k, u_1, \ell, u_2)\}$

Algorithm 4 COMPUTE_SPOLS

globals r, $<_T$
inputs
  P, a set of critical pairs in the form (t, k, u, ℓ, v)
outputs
  S, a list of indices in r of S-polynomials computed
  for a Gröbner basis of $(f_1, f_2, \ldots, f_i)$
do
  $S := ()$
  — Faugère and Stegers do not indicate that one should sort P, but performance suffers if not.
  — For the example in Faugère's paper, 8 polynomials would be computed, not 7.
  for $(t, k, u, \ell, v) \in P$, from smallest to largest lcm
    if not IS_REWRITABLE (u, k) **and not** IS_REWRITABLE (v, ℓ)
      Compute s, the S-polynomial of Poly (k) and Poly (ℓ)
      Append $(u \cdot \mathrm{Sig}(k), s)$ to r — Stegers writes $\mathrm{Sig}(\ell)$.
      ADD_RULE $(u \cdot \mathrm{Sig}(k), \#L)$
      if $s \neq 0$
        Append $\#r$ to S
  Sort S by increasing signature
  return S

Algorithm 5 REDUCTION

globals r, $<_T$
inputs
  S, a list of indices of polynomials added to the generators G_i
  B, a Gröbner basis of $(f_1, f_2, \ldots, f_{i-1})$ with respect to $<_T$
  $G_{\mathrm{prev}} \subset \mathbb{N}$, indices in r corresponding to B
  $G_{\mathrm{curr}} \subset \mathbb{N}$, indices in r of a list of generators of the ideal of $(f_1, f_2, \ldots, f_i)$
outputs
  completed, a subset of G corresponding to (mostly) top-reduced polynomials
do
  $\mathit{to\_do} := S$
  $\mathit{completed} := \emptyset$
  while $\mathit{to\_do} \neq ()$
    Let k be the element of $\mathit{to\_do}$ such that $\mathrm{Sig}(k)$ is minimal.
    $\mathit{to\_do} := \mathit{to\_do} \setminus \{k\}$
    $h := \mathrm{Normal\_Form}(\mathrm{Poly}(k), B, <_T)$
    $r_k := (\mathrm{Sig}(k), h)$
    $\mathit{newly\_completed}, \mathit{redo} :=$ TOP_REDUCTION $(k, G_{\mathrm{prev}}, G_{\mathrm{curr}} \cup \mathit{completed})$
    $\mathit{completed} := \mathit{completed} \cup \mathit{newly\_completed}$
    — Faugère and Stegers both write $\mathit{to\_do} := \mathit{to\_do} \cup \mathit{redo}$,
    — but $\mathit{to\_do}$ is not a set, and for efficiency needs to be sorted.
    for $j \in \mathit{redo}$
      Insert j in $\mathit{to\_do}$, sorting by increasing signature
  return *completed*

Algorithm 6 Top_Reduction

globals r, $<_T$
inputs
  k, the index of a labeled polynomial
  $G_{\text{prev}} \subset \mathbb{N}$, indices in r of a Gröbner basis of $(f_1, f_2, \ldots, f_{i-1})$ w/respect to $<_T$
  $G_{\text{curr}} \subset \mathbb{N}$, indices in r of a list of generators of the ideal of $(f_1, f_2, \ldots, f_i)$
outputs
  completed, which has value $\{k\}$ if r_k was **not** top-reduced and $\emptyset$ otherwise
  to_do, which has value
    $\emptyset$ if r_k was not top-reduced,
    $\{k\}$ if r_k is replaced by its top-reduction, and
    $\{k, \#r\}$ if top-reduction of r_k generates a polynomial with a signature larger than $\operatorname{Sig}(k)$.
do
  if $\operatorname{Poly}(k) = 0$ — This condition should be **false** if the inputs are a regular sequence.
    warn "Reduction to zero!"
    return $\emptyset, \emptyset$
  $p := \operatorname{Poly}(k)$
  $J := \operatorname{Find\_Reductor}(k, G_{\text{prev}}, G_{\text{curr}})$
  if $J = \emptyset$
    $r_k := \left(\operatorname{Sig}(k), p \cdot (\operatorname{lc}(p))^{-1}\right)$
    return $\{k\}, \emptyset$
  — $J \neq \emptyset$, so it is safe to top-reduce.
  Let j be the single element in J
  $q := \operatorname{Poly}(j)$
  $u := \frac{\operatorname{lt}(p)}{\operatorname{lt}(q)}$
  $c := \operatorname{lc}(p) \cdot (\operatorname{lc}(q))^{-1}$
  $p := p - c \cdot u \cdot q$
  if $p \neq 0$
    $p := p \cdot (\operatorname{lc}(p))^{-1}$
  if $u \cdot \operatorname{Sig}(j) \prec \operatorname{Sig}(k)$
    $r_k := (\operatorname{Sig}(k), p)$
    return $\emptyset, \{k\}$
  else
    Append $(u \cdot \operatorname{Sig}(j), p)$ to r
    $\text{Add\_Rule}(u \cdot \operatorname{Sig}(j), \#L)$
    — Faugère writes $\emptyset, \{k, j\}$ below, but $\operatorname{Poly}(\#L)$ needs top-reduction, not $\operatorname{Poly}(j)$.
    return $\emptyset, \{k, \#r\}$

Algorithm 7 Find_Reductor

globals $<_T$
inputs
  k, the index of a labeled polynomial
  $G_{\text{prev}} \subset \mathbb{N}$, indices in r of a Gröbner basis with respect to $<_T$ of $(f_1, f_2, \ldots, f_{i-1})$
  $G_{\text{curr}} \subset \mathbb{N}$, indices in r of a list of generators of the ideal of $(f_1, f_2, \ldots, f_i)$
outputs
  J, where $J = \{j\}$ if $j \in G_{\text{curr}}$ and $\text{Poly}(k)$ is *safely* top-reducible by $\text{Poly}(j)$;
    otherwise $J = \emptyset$
do
  $t := \text{lt}(\text{Poly}(k))$
  for $j \in G_{\text{curr}}$
    $t' = \text{lt}(\text{Poly}(j))$
    if $t' \mid t$
      $u := t/t'$
      $\mu_j \mathbf{F}_{\nu_j} := \text{Sig}(j)$
      if $u \cdot \text{Sig}(j) \neq \text{Sig}(k)$ **and not** Is_Rewritable(u, j) **and** $u \cdot \mu_j$ is not top-reducible by G_{prev}
        return $\{j\}$
  return $\emptyset$

Algorithm 8 Add_Rule

globals r, *Rule*
inputs
  $\mu \mathbf{F}_\nu$, the signature of r_k
  k, the index of a labeled polynomial in r (or 0, for a phantom labeled polynomial)
do
  Append (μ, k) to $Rule_\nu$
  return

Algorithm 9 Is_Rewritable

inputs
  u, a power product
  k, the index of a labeled polynomial in r
outputs
  true if $u \cdot \text{Sig}(k)$ is rewritable by another labeled polynomial (see Find_Rewriting)
do
  $j :=$ Find_Rewriting(u, k)
  return $j \neq k$

Algorithm 10 FIND_REWRITING

```
globals Rule
inputs
    u, a power product
    k, the index of a labeled polynomial in r
outputs
    j, the index of a labeled polynomial in r such that if μ_j F_{ν_j} = Sig(j)
        and μ_j F_{ν_j} = Sig(k), then ν_j = ν_k and μ_j | u · μ_k
        and r_j was added to Rule_{ν_k} most recently.
do
    μ_k F_ν := Sig(k)
    ctr := #Rule_ν
    while ctr > 0
        (μ_j, j) := Rule_{ν,ctr}
        if μ_j | u · μ_k
            return j
        ctr := ctr − 1
    return k
```