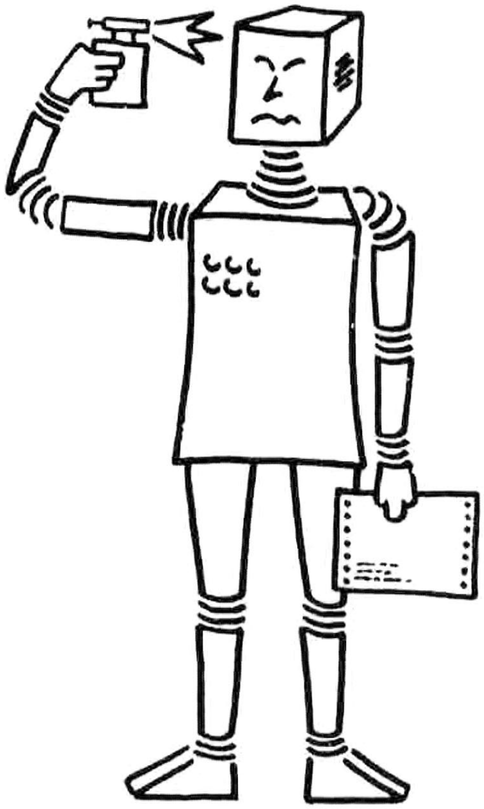
SEKI-PROJEKT

SEKI
MEMO

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern 1, W. Germany



SOLVING LINEAR DIOPHANTINEEQUATIONS

Thomas Guckenbiehl
Alexander Herold

MEMO SEKI-85-IV-KL

SOLVING LINEAR DIOPHANTINE EQUATIONS

Thomas Guckenbiehl
Alexander Herold

MEMO SEKI-85-IV-KL

SOLVING LINEAR DIOPHANTINE EQUATIONS

Thomas Guckenbiehl
Alexander Herold

Universität Kaiserslautern
Fachbereich Informatik
Postfach 3049
6750 Kaiserslautern

ABSTRACT:

Linear diophantine equations are at the heart of any unification algorithm for associative and commutative theories. The known algorithms for solving homogeneous linear diophantine equations of Huet and Fortenbacher are presented and their implementation is compared.
The algorithms for homogeneous equations are extended to solve inhomogeneous equations, since this is an important component of the AC-unification algorithm of Livesey and Siekmann and the extension of Herold and Siekmann.

CONTENTS

1. INTRODUCTION

The interest in diophantine equations has a long tradition in mathematics. However, recently algorithms to solve such equations became of great practical relevance in computer-science, playing a central role in unification-algorithms for terms with associative and commutative function-symbols (for short: AC-Unification). AC-Unification turned out to be of practical importance for term rewriting systems, automated theorem proving and many programming languages in Artificial Intelligence (AI).

In the literature two different approaches to the AC-Unification-Problem are known: one based on M. Stickel [St 75], [St 81] and the other based on Livesey and Siekmann [LS 76], which recently was extended by Herold and Siekmann [HS 85]. A theoretical comparison is found in [Bü 85]. The most important difference between the two algorithms is the reduction of the problem to linear diophantine equations: Stickel abstracts his AC-Unification problem to a pure variable unification problem which leads to a homogeneous equation, whereas Herold and Siekmann directly determine one homogeneous equation, which is smaller than Stickel's, and a system of inhomogeneous equations with the same homogeneous part.

For example, Stickel [St 81] gives the problem of unifying the terms $f(x, f(x, f(y, a)))$ and $f(b, f(b, z))$, where f is an associative and commutative function-symbol, and this yields the homogeneous linear diophantine equation $2x_1 + x_2 + x_3 = y_1 + 2y_2$. The same example is treated in [HS 85] . There the reduction yields the equations

$$\begin{aligned} 2x_{11} + x_{12} &= y_{11} \\ 2x_{21} + x_{22} + 1 &= y_{21} \\ 2x_{31} + x_{32} &= y_{31} + 2 . \end{aligned}$$

The set of all solutions with non-negative integer coefficients of a homogeneous linear diophantine equation forms a commutative monoid which is finitely generated [CP 67]. Sequential algorithms for determining such a generator-set have been proposed by Huet [Hu 78], Fortenbacher [Fo 83] and Lankford [La 85], a parallel algorithm was outlined by Büttner [Bü 85].

After an introduction to the theory of linear diophantine equations we present the algorithms of Huet and Fortenbacher and extend them to compute the sets of all minimal solutions of a system of inhomogeneous equations like the one in the example above. The main difference between both is that Huet's algorithm is essentially depth-first whereas Fortenbacher's is breadth-first. The algorithms have been implemented in ZETA-LISP on a SYMBOLICS 3640, a comparison in terms of the runtime on various examples is presented.

The most important result is that in most cases it is faster to solve the multiple equations of the Livesey and Siekmann algorithm than the larger homogeneous equation of Stickel.

2. SOLVING LINEAR DIOPHANTINE EQUATIONS OVER $\mathbf{N}$

2.1 The Concept of Minimality

We want to introduce some notion and definitions about vectors of non-negative integers:

Let $\mathbf{N}_0$ be the set of non-negative integers, $\mathbf{N} = \mathbf{N}_0\backslash\{0\}$ the set of positive integers, and let $x_1, \dots, x_m$ be elements in $\mathbf{N}_0^n$. A ***p-linear combination*** (positive linear combination) of the x_i is

$$y = a_1 x_1 + \dots + a_m x_m \text{ , with } a_i \geq 0$$

Vectors are called ***p-independent,*** iff none of them can be represented as a p-linear combination of the others. If $S \subseteq \mathbf{N}_0^n\backslash\{0\}$ and $M \subseteq S$, then M is a ***p-basis*** of S, iff the elements of M are p-independent and every element of S can be represented as a p-linear combination of elements of M.

Let $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ be in $\mathbf{N}_0^n$. We define a strict order

$$x \ll y \qquad \text{iff} \qquad x \neq y \ \underline{\text{and}} \ x_i \leq y_i \text{ for } 1 \leq i \leq n .$$

Let S be a set of vectors in $\mathbf{N}_0^n\backslash\{0\}$, then $x \in S$ is ***minimal in*** S, iff there is no $y \in S$ with $y \ll x$.

LEMMA 2-1: Let $S \subseteq \mathbf{N}_0^n\backslash\{0\}$. If every element of S is minimal in S, then the elements are p-independent.

<u>PROOF</u>: Assume that S is minimal and there exists $x \in S$ and $M \subset S\backslash\{x\}, M \neq \emptyset$, such that $x = \sum_{y \in M} a_y y$, with $a_y > 0$. Then every element of M must be smaller than x.

But this is a contradiction to the minimality of x.

■

To illustrate the notions we shall give some examples:

First we have $(1, 2, 1) \ll (2, 2, 2) \ll (2, 3, 3)$,

but <u>not</u> $(1, 2, 1) \ll (2, 1, 2)$

and <u>not</u> $(2, 1, 2) \ll (1, 2, 1)$.

The set $M = \{(2, 1, 1), (1, 2, 1), (1, 1, 2), (1, 0\ 3)\}$ is minimal and its elements are p-independent.

The elements of $M' = \{(1, 1, 1), (1, 2, 1)\}$ are p-independent, but M' is <u>not</u> minimal!

2.2 Homogeneous Linear Diophantine Equations

Let $\mathfrak{a} = (a_1, \dots, a_m) \in \mathbf{N}^m$, $\mathfrak{b} = (b_1, \dots, b_n) \in \mathbf{N}^n$ and n, m $\geq$ 0, then $\mathfrak{a}$ and $\mathfrak{b}$ define a homogeneous linear diophantine equation

$$\mathrm{HOM}(\mathfrak{a},\mathfrak{b}): \; a_1 x_1 + \dots + a_m x_m = b_1 y_1 + \dots + b_n y_n .$$

The set of all solutions $\mathbf{u} \in \mathbf{N}_0^{m+n} \backslash \{\mathbf{0}\}$ is denoted by $S(\mathfrak{a}, \mathfrak{b})$, the set of all minimal solutions in $S(\mathfrak{a}, \mathfrak{b})$ is denoted by $M(\mathfrak{a}, \mathfrak{b})$. (Note that we only want to consider nontrivial solutions !)

We do not consider the trivial case that both $n = 0$ and $m = 0$, i. e. the trivial equation $0 = 0$. If $n = 0$ or $m = 0$, we define $\mathrm{HOM}(\mathfrak{a}, \mathfrak{b})$ as $a_1 x_1 + \dots + a_m x_m = 0$ resp. $0 = b_1 y_1 + \dots + b_n y_n$.

For example, given $\mathfrak{a} = (2, 1) \in \mathbf{N}^2$ and $\mathfrak{b} = (1, 1, 2) \in \mathbf{N}^3$, we have

$$\mathrm{HOM}(\mathfrak{a}, \mathfrak{b}): 2 x_1 + x_2 = y_1 + y_2 + 2 y_3$$

and $M(\mathfrak{a}, \mathfrak{b}) = \{(1, 0, 0, 0, 1), (1, 0, 1, 1, 0), (0, 1, 1, 0, 0), (0, 1, 0, 1, 0),$
$(0, 2, 0, 0, 1), (1, 0, 2, 0, 0), (1, 0, 0, 2, 0)\}$.

Since we are only interested in linear diophantine equations, we shall drop "linear" and "diophantine" and just write "homogeneous equation".

We shall now explore, under which conditions there exist solutions to a homogeneous equation and then take a closer look at the sets $S(\mathfrak{a}, \mathfrak{b})$ and $M(\mathfrak{a}, \mathfrak{b})$.

LEMMA 2-2 : Let $\mathfrak{a}$, $\mathfrak{b}$ define a homogeneous equation.
There exists a (nontrivial) solution iff $n > 0$ <u>and</u> $m > 0$.

<u>PROOF :</u> "$\Rightarrow$": Assume $S(\mathfrak{a}, \mathfrak{b}) \neq \emptyset$ and w. l. o. g. $n = 0$. Let $s = (c_1, \dots, c_m)$ be a nontrivial solution. If W. l. o. g. $c_1 > 0$, we have

$$0 = a_1 c_1 + \dots + a_m c_m \geq a_1 c_1 > 0 \quad \lightning.$$

"$\Leftarrow$": Let n, m > 0 and consider $s = (b_1, c_2, \dots, c_m, a_1, d_2, \dots, d_n)$, with $c_i = 0$ for $2 \leq i \leq m$ and $d_j = 0$ for $2 \leq j \leq n$. Then s is obviously a solution.

■

LEMMA 2-3 : Let $\mathfrak{a}$ and $\mathfrak{b}$ define a homogeneous equation and let $g := \gcd(a_1, \ldots, a_m, b_1, \ldots, b_n)$.
Define $\mathfrak{a}' = (a_1', \ldots, a_m')$, $\mathfrak{b}' = (b_1', \ldots, b_n')$ by

$$a_i' = (1/g)\ a_i \quad \text{for } 1 \le i \le m$$
$$b_j' = (1/g)\ b_j \quad \text{for } 1 \le j \le n \quad .$$

Then $S(\mathfrak{a}, \mathfrak{b}) = S(\mathfrak{a}', \mathfrak{b}')$.

PROOF : Obviously $\mathfrak{a}' \in \mathbf{N}^m$, $\mathfrak{b}' \in \mathbf{N}^n$.
Now $s = (c_1, \ldots, c_m, d_1, \ldots, d_n) \in S(\mathfrak{a}, \mathfrak{b})$

$$\begin{aligned}
\Leftrightarrow \quad & a_1c_1 + \ldots + a_mc_m &&= b_1d_1 + \ldots + b_nd_n \\
\Leftrightarrow \quad & (1/g)\,(a_1c_1 + \ldots + a_mc_m) &&= (1/g)\,(b_1d_1 + \ldots + b_nd_n) \\
\Leftrightarrow \quad & a_1'\,c_1 + \ldots + a_m'\,c_m &&= b_1'\,d_1 + \ldots + b_n'\,d_n \\
\Leftrightarrow \quad & s \in S(\mathfrak{a}', \mathfrak{b}')
\end{aligned}$$

■

Because of the last two lemmas we may restrict ourselves to homogeneous equations with n, m > 0 and $\gcd(a_1, \ldots, a_m, b_1, \ldots, b_n) = 1$.
Since the set of all solutions $S(\mathfrak{a}, \mathfrak{b})$ forms a commutative submonoid of $\mathbf{N}_0^{n+m}$, it is generated by a finite basis (c. f. [CP 67]), which consists of all minimal elements of $S(\mathfrak{a}, \mathfrak{b})$. Hence $M(\mathfrak{a}, \mathfrak{b})$ is a finite p-basis of $S(\mathfrak{a}, \mathfrak{b})$.

THEOREM 2-4 : Let $\mathfrak{a} = (a_1, \ldots, a_m) \in \mathbf{N}^m$ and $\mathfrak{b} = (b_1, \ldots, b_n) \in \mathbf{N}^n$ define a homogeneous equation. Define $\max_a := \max(a_1, \ldots, a_m\}$ and $\max_b := \max(b_1, \ldots, b_n\}$. Then the components of every minimal solution $s = (c_1, \ldots, c_m, d_1, \ldots, d_n)$ are bounded by $0 \le c_i \le \max_b$ and $0 \le d_j \le \max_a$ for $1 \le i \le m$ and $1 \le j \le n$.

For a proof see [Hu 78].
Since this theorem defines a finite and effectively computable search-space, we can compute $M(\mathfrak{a}, \mathfrak{b})$ by generating each element of this space and removing those that are no minimal solutions. In fact this is the main idea of Huet's algorithm [Hu 78], but he uses additional bounds to reduce the generation of p-dependent elements.

2.3 Inhomogeneous Linear Diophantine Equations

Let $m, n \geq 0$, $\mathfrak{a} = (a_1, \dots, a_m) \in \mathbf{N}^m$, $\mathfrak{b} = (b_1, \dots, b_n) \in \mathbf{N}^n$, and $P = \{p_1, \dots, p_r\} \subset \mathbf{Z}$. Then $\mathfrak{a}$, $\mathfrak{b}$ and P define a system of r inhomogeneous linear diophantine equations

$$\mathrm{INHOM}(\mathfrak{a}, \mathfrak{b}, P): \quad \begin{array}{rcl} a_1x_{11} + \dots + a_mx_{1m} & = & b_1y_{11} + \dots + b_ny_{1n} \;+\; p_1 \\ \dots & & \dots \\ a_1x_{r1} + \dots + a_mx_{rm} & = & b_1y_{r1} + \dots + b_ny_{rn} \;+\; p_r \end{array}$$

In this chapter we only deal with a single inhomogeneous equation, i. e. $P = \{p\}$ with $p \in \mathbf{Z}$. Instead of $\mathrm{INHOM}(\mathfrak{a}, \mathfrak{b}, P)$ we shall therefore write $\mathrm{INHOM}_p(\mathfrak{a}, \mathfrak{b})$. Notice that we allow $p = 0$ and treat the homogeneous equation as a special case.

As with homogeneous equations we do not want to consider the trivial case that both $n = 0$ and $m = 0$. If $n = 0$ or $m = 0$ we define $\mathrm{INHOM}_p(\mathfrak{a}, \mathfrak{b})$ as $a_1x_1 + \dots + a_mx_m = p$ and $0 = b_1y_1 + \dots + b_ny_n + p$, respectively.

The set of all (nontrivial) solutions $\mathfrak{u} \in \mathbf{N}_0^{m+n} \backslash \{\mathbf{0}\}$ to $\mathrm{INHOM}_p(\mathfrak{a}, \mathfrak{b})$ is denoted by $S_p(\mathfrak{a}, \mathfrak{b})$, the set of all minimal solutions is $M_p(\mathfrak{a}, \mathfrak{b}) \subset S_p(\mathfrak{a}, \mathfrak{b})$.

As with homogeneous equations, we shall drop "linear" and "diophantine" and just write "inhomogeneous equation".

First we shall look for conditions under which the set of solutions is empty:

Theorem 2-5 : Let $\mathfrak{a}$, $\mathfrak{b}$ and p define an inhomogeneous equation INHOM_p and define $g := \gcd(a_1, \dots, a_m, b_1, \dots, b_n)$.

(1) If $n > 0$ <u>and</u> $m > 0$, then
there exists a solution iff $g \mid p$.

(2) If $n = 0$ <u>or</u> $m = 0$ then we have :
(a) If there is a solution then $g \mid p$.
(b) If $m = 0$ <u>and</u> $p \geq 0$ then there is no solution .
(c) If $n = 0$ <u>and</u> $p \leq 0$ then there is no solution .

<u>PROOF :</u> <u>Ad (1):</u> "$\Rightarrow$": If $s = (c_1, \dots, c_m, d_1, \dots, d_n)$ is a nontrivial solution to INHOM_p, we have $a_1c_1 + \dots + a_mc_m = b_1d_1 + \dots + b_nd_n + p$. Since all coefficients a_i and b_j are from $g\mathbf{Z}$ and since $g\mathbf{Z}$ is an ideal of $\mathbf{Z}$, $p \in g\mathbf{Z}$, i. e. $g \mid p$ follows.
"$\Leftarrow$": Let $p/g \in \mathbf{Z}$. For $1 \leq i \leq m$ and $1 \leq j \leq n$ define $s_{ij} := (c_1, \dots, c_m, d_1, \dots, d_n)$ where $c_k = b_j$ for $k = i$ and $c_k = 0$ for $1 \leq k \neq i \leq m$, $d_k = a_i$ for $k = j$ and $d_k = 0$

for $1 \le k + j \le n$. These vectors are solutions of the homogeneous equation. Furthermore we know (from number theory) that the greatest common divisor g is linear representable, i. e. there is a solution $\mathfrak{u} = (u_1, \dots, u_m, v_1, \dots, v_n) \in \mathbb{Z}^{m+n}$ with $a_1 u_1 + \dots + a_m u_m = b_1 v_1 + \dots + b_n v_n + g$. Multiplying by $c := p/g$ we get $a_1 c u_1 + \dots + a_m c u_m = b_1 c v_1 + \dots + b_n c v_n + p$.

Now suppose there is i or j with $u_i < 0$ or $v_j < 0$. Then by adding s_{ij} to $\mathfrak{u}$ for all these i, j we obtain a solution with non-negative components.

Ad (2): (2b) and (2c) are obvious. Our proof of the first part of (1) is also a proof for (2a).

■

Lemma 2-6 : Let $m, n > 0$. Let $\mathfrak{a} \in \mathbb{N}^n$, $\mathfrak{b} \in \mathbb{N}^m$ and p define an inhomogeneous equation INHOM_p.
Define $g := \gcd(a_1, \dots, a_m, b_1, \dots, b_n)$,
$\mathfrak{a}' := (a_1', \dots, a_m')$ with $a_i' = (1/g)\, a_i$ for $1 \le i \le m$ and
$\mathfrak{b}' := (b_1', \dots, b_n')$ with $b_j' = (1/g)\, b_j$ for $1 \le j \le n$.
If $g \mid p$ then it is $S_p(\mathfrak{a}, \mathfrak{b}) = S_{p/g}(\mathfrak{a}', \mathfrak{b}')$.

PROOF : $s = (c_1, \dots, c_m, d_1, \dots, d_n) \in S_p(\mathfrak{a}, \mathfrak{b})$

$$\begin{aligned} &\Leftrightarrow a_1 c_1 + \dots + a_m c_m = b_1 d_1 + \dots + b_n d_n + p \\ &\Leftrightarrow a_1' c_1 + \dots + a_m' c_m = b_1' d_1 + \dots + b_n' d_n + p/g \\ &\Leftrightarrow s \in S_{p/g}(\mathfrak{a}', \mathfrak{b}'). \end{aligned}$$

■

From now on we shall only consider inhomogeneous equations with $n, m > 0$ and $\gcd(a_1, \dots, a_m, b_1, \dots, b_n) = 1$, which, according to Theorem 2-5, always have a solution.

Theorem 2-7 : The set of positive integer solutions $S_p(\mathfrak{a}, \mathfrak{b})$ of an inhomogeneous linear diophantine equation INHOM_p is
$\{ \mathfrak{y} \mid \mathfrak{y} = \mathfrak{x}_p + \mathfrak{x}_0 \text{ with } \mathfrak{x}_p \in M_p(\mathfrak{a}, \mathfrak{b}) \text{ and } \mathfrak{x}_0 \in S(\mathfrak{a}, \mathfrak{b}) \cup \{\mathbf{0}\} \}$.

Proof: It is easy to see that $\mathfrak{x}_p + \mathfrak{x}_0$ is in $S_p(\mathfrak{a}, \mathfrak{b})$. Conversely let $\mathfrak{y} \in S_p(\mathfrak{a}, \mathfrak{b})$ then if $\mathfrak{y}$ is in $M_p(\mathfrak{a}, \mathfrak{b})$ we are done with $\mathfrak{x} = \mathbf{0}$. Suppose $\mathfrak{y}$ is not in $M_p(\mathfrak{a}, \mathfrak{b})$ then there exists by definition $\mathfrak{x}_p$ in $M_p(\mathfrak{a}, \mathfrak{b})$ with $\mathfrak{x}_p < \mathfrak{y}$. Therefore we have $\mathfrak{y} - \mathfrak{x}_p > \mathbf{0}$ and hence $\mathfrak{y} - \mathfrak{x}_p = \mathfrak{x}_0 \in S(\mathfrak{a}, \mathfrak{b})$, i. e. $\mathfrak{y} = \mathfrak{x}_p + \mathfrak{x}_0$. ■

We now state an important relation between the sets $M_p(\mathfrak{a}, \mathfrak{b})$ and $M(\mathfrak{a}, \mathfrak{b})$, i. e. the minimal solutions of an inhomogeneous equation and those of the corresponding homogeneous equation.

THEOREM 2-8 : Let $\mathrm{INHOM}_p(\mathfrak{a}, \mathfrak{b})$ be an inhomogeneous equation. $s \in S_p(\mathfrak{a}, \mathfrak{b})$ is minimal in $S_p(\mathfrak{a}, \mathfrak{b})$ iff there is no $\boldsymbol{v} \in M(\mathfrak{a}, \mathfrak{b})$ with $\boldsymbol{v} \ll s$.

<u>PROOF :</u> Let s be a solution to the inhomogeneous equation.
First let s be minimal in $S_p(\mathfrak{a}, \mathfrak{b})$ and assume there is $\boldsymbol{v} \in M(\mathfrak{a}, \mathfrak{b})$ with $\boldsymbol{v} \ll s$. But then $s - \boldsymbol{v} \ll s$ is also a non-negative solution of the inhomogeneous equation, a contradiction to the minimality of s.
Now assume there is no $\boldsymbol{v} \in M(\mathfrak{a}, \mathfrak{b})$ with $\boldsymbol{v} \ll s$, and s is not minimal in $S_p(\mathfrak{a}, \mathfrak{b})$. Hence there exists $\boldsymbol{w} \in S_p(\mathfrak{a}, \mathfrak{b})$ with $\boldsymbol{w} \ll s$, and $s - \boldsymbol{w} \ll s$ is in $S(\mathfrak{a}, \mathfrak{b})$. ↯

■

As proved in [LS 76], $M_p(\mathfrak{a}, \mathfrak{b})$ is bounded. Maiwand [Mai 78] showed, that every component of a minimal solution to $\mathrm{INHOM}_p(\mathfrak{a}, \mathfrak{b})$ must be smaller than $\max\{a_1, \dots, a_m, b_1, \dots, b_n, p\}$. In Theorem 2-10 we shall give a slightly better bound, for which we need the following technical lemma :

LEMMA 2-9 : Let $\mathfrak{a}$, $\mathfrak{b}$ and p define an inhomogeneous equation INHOM_p.

(1) If $p > 0$, define $\mathfrak{b}' := (b_1, \dots, b_n, p)$. Then
$$M_p(\mathfrak{a}, \mathfrak{b}) = \{(c_1, \dots, c_m, d_1, \dots, d_n) \mid (c_1, \dots, c_m, d_1, \dots, d_n, 1) \in M(\mathfrak{a}, \mathfrak{b}')\}.$$

(2) If $p < 0$, define $\mathfrak{a}' := (a_1, \dots, a_m, -p)$. Then
$$M_p(\mathfrak{a}, \mathfrak{b}) = \{(c_1, \dots, c_m, d_1, \dots, d_n) \mid (c_1, \dots, c_m, 1, d_1, \dots, d_n) \in M(\mathfrak{a}', \mathfrak{b})\}.$$

<u>PROOF :</u> We only prove (1); (2) follows by exchanging $\mathfrak{a}$ and $\mathfrak{b}$.
"$\subseteq$": If $\mathfrak{u} = (u_1, \dots, u_{m+n})$ is a minimal solution of $\mathrm{INHOM}_p(\mathfrak{a}, \mathfrak{b})$, then obviously $\mathfrak{u}' = (u_1, \dots, u_{m+n}, 1)$ is a solution to $\mathrm{HOM}(\mathfrak{a}, \mathfrak{b}')$. Now assume there is $\boldsymbol{v} = (v_1, \dots, v_{m+n+1}) \in M(\mathfrak{a}, \mathfrak{b}')$ and $\boldsymbol{v} \ll \mathfrak{u}'$. If $v_{m+n+1} = 1$, then $(v_1, \dots, v_{m+n}) \ll \mathfrak{u}$ is a solution of $\mathrm{INHOM}_p(\mathfrak{a}, \mathfrak{b})$. If $v_{m+n+1} = 0$, then $(u_1 - v_1, \dots, u_{m+n} - v_{m+n}) \ll \mathfrak{u}$ is a solution of $\mathrm{INHOM}_p(\mathfrak{a}, \mathfrak{b})$. In both cases we get a contradiction to the minimality of $\mathfrak{u}$.

"$\supseteq$": Let $\mathfrak{u}' = (u_1, \ldots, u_{m+n}, 1)$ be a minimal solution of HOM($\mathfrak{a}$, $\mathfrak{b}'$). Then obviously $\mathfrak{u} = (u_1, \ldots, u_{m+n})$ is a solution of INHOM$_p$($\mathfrak{a}$, $\mathfrak{b}$). Now assume there is $\mathfrak{v} = (v_1, \ldots, v_{m+n}) \in M_p(\mathfrak{a}, \mathfrak{b})$ and $\mathfrak{v} \ll \mathfrak{u}$. But then $(v_1, \ldots, v_{m+n}, 1) \ll \mathfrak{u}'$ is a solution of HOM($\mathfrak{a}$, $\mathfrak{b}'$) and therefore $\mathfrak{u}'$ could not be minimal.

■

THEOREM 2-10 : Let $\mathfrak{a}$, $\mathfrak{b}$ and p define an inhomogeneous equation INHOM$_p$
Define $\max_1 := \max\{a_1, \ldots, a_m, -p\}$
$\max_2 := \max\{b_1, \ldots, b_n, p\}$.
Then the components of every minimal solution $\mathfrak{s} = (c_1, \ldots, c_m, d_1, \ldots, d_n)$ to INHOM$_p$($\mathfrak{a}$, $\mathfrak{b}$) are bound by $0 \leq c_i \leq \max_2$ and $0 \leq d_j \leq \max_1$ for $1 \leq i \leq m$ and $1 \leq j \leq n$.

The proof follows from Lemma 2-9 by applying the bounds of Theorem 2-4 to the extended homogeneous equation.
As mentioned in the introduction, we are interested in a system of multiple inhomogeneous equations and one corresponding homogeneous equation. With respect to AC-Unification we expect that the inhomogeneous parts p will not be substantially larger than the coefficients a_i, b_j. But then, according to Theorems 2-4 and 2-10, the search-spaces are overlapping. So we should avoid generating the same elements for different equations, but try to test each generated vector, if it is a solution to anyone of them .
In the next two chapters we shall use this idea to extend the algorithms of Huet and Fortenbacher from homogeneous to systems of inhomogeneous equations. Theorem 2-8 will help us in doing so.

3 THE ALGORITHM OF HUET

3.1 The Homogeneous Case

In 1978 Gérard Huet described an algorithm to solve homogeneous linear diophantine equations over $\mathbf{N}_0$ [Hu 78]. Before presenting it, we have to do some preliminaries:

Let $\mathfrak{a} = (a_1 ,..., a_m) \in \mathbf{N}^m$ and $\mathfrak{b} = (b_1 ,..., b_n) \in \mathbf{N}^n$ define a homogeneous equation HOM($\mathfrak{a}, \mathfrak{b}$); then let

$$\begin{aligned} max_a &:= max\{a_1 ,..., a_m\} \\ max_b &:= max\{b_1 ,..., b_n\}. \end{aligned}$$

For $1 \le i \le m$ and $1 \le j \le n$ we define

$d_{ij} := lcm(a_i, b_j) / a_i$ (least common multiple)

$e_{ij} := lcm(a_i, b_j) / b_j$

$s_{ij} := (x_1 ,..., x_m, y_1 ,..., y_n) \in \mathbf{N}_0^{m+n}$, with $x_i = d_{ij}$, $y_j = e_{ij}$, all other components 0.

$maxy_j^i := min\{e_{tj} - 1 \mid x_t \ge d_{tj} , 1 \le t \le i\}$, if this set is not empty, max_a otherwise.

The s_{ij} are obviously special minimal solutions. Now Huet states some important properties of minimal solutions of HOM($\mathfrak{a}, \mathfrak{b}$):

THEOREM 3-1: Let $\mathfrak{a}$ and $\mathfrak{b}$ define a homogeneous equation and let $\mathfrak{u} = (x_1 ,..., x_m, y_1 ,..., y_n)$ be a minimal solution to HOM($\mathfrak{a}, \mathfrak{b}$), different from the s_{ij}.

Then for $1 \le k \le m$

(a) $x_i \le max_b$ for $1 \le i \le k$

(b) $a_1 x_1 + ... + a_k x_k \le b_1\ maxy_1^k + ... + b_n\ maxy_n^k$.

And for $1 \le s \le n$

(c) $y_j \le maxy_j^m$ for $1 \le j \le s$

(d) $b_1 y_1 + ... + b_s y_s \le a_1 x_1 + ... + a_m x_m$.

For the proof see [Hu 78].
Now we can cite Huet's description of the algorithm ([Hu 78], pp 145 f), with minor notational adjustments:

> Our algorithm consists in generating potential solutions in increasing lexicographic order, starting with (and not including) the trivial solution $(0, \dots, 0)$. The values of the x_i's are progressively bounded according to the conditions (a) and (b) above. When all x_i's are chosen, the y_i's are bounded according to the conditions (c) and (d).
> When the algorithm generates a potential solution (x, y), it checks that
> (1) it is indeed a solution (...) [to HOM($\mathfrak{a}$, $\mathfrak{b}$)]
> (2) it is not greater than any solution previously generated.
> Then it backtracks to generate further solutions. When it finally stops, all particular solutions s_{ij}'s are added. Remark that when a solution is generated and checked, it is indeed minimal, because any solution generated later on will either be greater in the lexicographic order (and therefore not below it in the componentwise order), or one of the s_{ij}'s, which cannot be below it by construction. The s_{ij}'s are themselves known to be minimal, which finishes the correctness proof of our algorithm.

In the following we shall call these potential solutions, generated by the algorithm, ***proposals***.
In addition to these upper bounds for the y_j, one might use the following theorem which gives a lower bound for them.

THEOREM3-2 : Let $\mathfrak{a}$ and $\mathfrak{b}$ define a homogeneous equation. Then for every minimal solution $\mathbf{u} = (x_1, \dots, x_m, y_1, \dots, y_n)$ and for $1 \le i \le n$ we have $y_i \ge \max\{0, c_i/b_i\}$, with

$$c_i := \sum_{j=1}^{m} a_j x_j - \sum_{j=1}^{i-1} b_j y_j - \sum_{j=i+1}^{n} b_j \, \mathrm{maxy}_j{}^m.$$

PROOF : Since $\mathbf{u}$ is a solution, it is for $1 \le i \le n$:

$$\sum_{j=1}^{m} a_j x_j = \sum_{j=1}^{i-1} b_j y_j + b_i y_i + \sum_{j=i+1}^{n} b_j y_j$$

and therefore
$$b_i y_i = \sum_{j=1}^{m} a_j x_j - \sum_{j=1}^{i-1} b_j y_j - \sum_{j=i+1}^{n} b_j y_j$$

With Theorem 3-1 (c) we now get $b_i y_i \ge c_i$.

■

3.2 The Inhomogeneous Case

We shall now show, how Huet's algorithm for homogeneous equations can be extended to find in parallel the minimal solutions of a system of inhomogeneous equations as defined in 2.3.
The resulting algorithm will be essentially the same as Huet's, but will use four slightly different bounding criteria. After generating a proposal, we additionally have to test, if it is a solution not only of the homogeneous equation, but of any of the equations. Theorem 2-8 yields that no solution is lost, if we test the minimality of a proposal by comparing it with the minimal solutions of the homogeneous equation.

If $\mathfrak{a}$, $\mathfrak{b}$ and $P = \{p_1, \dots, p_r\} \subset \mathbb{Z}$ give a system of inhomogeneous equations INHOM($\mathfrak{a}$, $\mathfrak{b}$, P), we define d_{ij}, e_{ij} and s_{ij} as for the homogeneous case. But we replace the definitions of $\max_a$, $\max_b$ and $\text{maxy}_j{}^i$ by

$\max_1 \quad := \max(\{a_1, \dots, a_m\} \cup P^-)$, where $P^- = \{-p \mid p \in P\}$

$\max_2 \quad := \max(\{b_1, \dots, b_n\} \cup P)$

$\text{maxy-inh}_j{}^i \quad := \min\{e_{tj} - 1 \mid x_t \geq d_{tj},\ 1 \leq t \leq i\}$, if this set is not empty,

$\qquad\qquad \max_1$ otherwise.

And in addition we define

$\max_p \quad := \max(P \cup \{0\})$

$\min_p \quad := \min(P \cup \{0\})$.

Now we can extend Huet's criteria to our inhomogeneous problem :

THEOREM 3-3: Let $\mathfrak{a}$, $\mathfrak{b}$ and P define a system of inhomogeneous equations INHOM($\mathfrak{a}$, $\mathfrak{b}$, P) and let $\mathfrak{u} = (x_1, \dots, x_m, y_1, \dots, y_n)$ be a minimal solution to $\text{INHOM}_p(\mathfrak{a}, \mathfrak{b})$ for some $p \in P$, but different from every s_{ij}.

Then for $1 \leq k \leq m$

(a) $x_i \leq \max_2$ for $1 \leq i \leq k$

(b) $a_1 x_1 + \dots + a_k x_k \leq b_1 \text{ maxy-inh}_1{}^k + \dots + b_n \text{ maxy-inh}_n{}^k + \max_p$.

And for $1 \leq s \leq n$

(c) $y_j \leq \text{maxy-inh}_j{}^m$ for $1 \leq j \leq s$

(d) $b_1 y_1 + \dots + b_s y_s + \min_p \leq a_1 x_1 + \dots + a_m x_m$.

PROOF : Ad (a) : follows from Theorem 2-10.

Ad (b):
$$\begin{aligned} a_1 x_1 + \dots + a_k x_k &\leq a_1 x_1 + \dots + a_m x_m \\ &= b_1 y_1 + \dots + b_n y_n + p \\ &\leq b_1 y_1 + \dots + b_n y_n + max_p \\ \text{because of (c):} &\leq b_1 \, \text{maxy-inh}_1^m + \dots + b_n \, \text{maxy-inh}_n^m + max_p \\ &\leq b_1 \, \text{maxy-inh}_1^k + \dots + b_n \, \text{maxy-inh}_n^k + max_p \end{aligned}$$

Ad (c): If there is no t with $x_t \geq d_{tj}$, then $y_j > \text{maxy-inh}_j^m = max_1$ is a contradiction to Theorem 2-10. However, if there is $t \leq k$ with $x_t \geq d_{tj}$ and $\text{maxy-inh}_j^m = e_{tj} - 1$, then $y_j \geq e_{tj}$ means $s_{tj} \ll u$, since $u \neq s_{tj}$. But s_{tj} is a solution to the homogeneous equation, which contradicts the minimality of **u** by Theorem 2-8 .

Ad (d): Since **u** is a solution to $INHOM_p(\mathfrak{a}, \mathfrak{b})$,we have

$$\begin{aligned} b_1 y_1 + \dots + b_s y_s + min_p &\leq b_1 y_1 + \dots + b_n y_n + p \\ &= a_1 x_1 + \dots + a_m x_m \, . \end{aligned}$$

■

In addition to these bounding-criteria for the y_j's , there is another bound, analogous to that of Theorem 3-2 :

THEOREM 3-4: Let $\mathfrak{a}, \mathfrak{b}$ and P define a system of inhomogeneous equations. Then for every minimal solution $u = (x_1, \dots, x_m, y_1, \dots, y_n)$ of any of the equations and for all $i \leq n$, the y_i are bounded below by $y_i \geq \max\{0, c_i/b_i\}$, with

$$c_i := \sum_{j=1}^{m} a_j x_j - \sum_{j=1}^{i-1} b_j y_j - \sum_{j=i+1}^{n} b_j \, \text{maxy-inh}_j^m - max_p.$$

PROOF : If $u \in M_p(\mathfrak{a}, \mathfrak{b})$ for some $p \in P$, we have for $1 \leq i \leq n$:

$$\sum_{j=1}^{m} a_j x_j = \sum_{j=1}^{i-1} b_j y_j + b_i y_i + \sum_{j=i+1}^{n} b_j y_j + p$$

and therefore
$$b_i y_i = \sum_{j=1}^{m} a_j x_j - \sum_{j=1}^{i-1} b_j y_j - \sum_{j=i+1}^{n} b_j y_j - p \, .$$

With Theorem 3-3 (c) and $p \leq max_p$ we now get $b_i y_i \geq c_i$.

■

4 : THE ALGORITHM OF FORTENBACHER

4.1 The Homogeneous Case

As we have seen in chapter 3, Huet's algorithm generates proposals in a lexical order. This might be regarded as a depth-first search in the grid $[0 ,\ldots, max_{\mathfrak{b}}] \times [0 ,\ldots, max_{\mathfrak{a}}]$. In contrast, Fortenbacher presents a kind of breadth-first algorithm [Fo 83], for the description of which we need two concepts:

For a vector $\boldsymbol{u} = (u_1 ,\ldots, u_m, v_1 ,\ldots, v_n) \in \mathbf{N}_0^{m+n}$ we define the ***cross-sum*** $s(\boldsymbol{u}) := u_1 + \ldots + u_m + v_1 + \ldots + v_n$. Obviously, if $\boldsymbol{w} \ll \boldsymbol{u}$ then $s(\boldsymbol{w}) < s(\boldsymbol{u})$. Furthermore we define the ***difference*** between the left-hand and the right-hand side of the homogeneous equation for $\boldsymbol{u}$:

$$d(\boldsymbol{u}) := (a_1 u_1 + \ldots + a_m u_m) - (b_1 v_1 + \ldots + b_n v_n).$$

Fortenbacher starts with the set L^1 of those $(m+n)$ proposals $\boldsymbol{u} \in \mathbf{N}_0^{n+m}$, which have $s(\boldsymbol{u}) = 1$. In the k-th step, $k \geq 1$, he constructs two sets L^{k+1} and $M^k := \{s \in M(\mathfrak{a}, \mathfrak{b}) \mid s(s) < k\}$ from those elements $\boldsymbol{u} = (u_1 ,\ldots, u_m , v_1 ,\ldots, v_n)$ of L^k, for which there is no $\boldsymbol{v} \in M^k$ with $\boldsymbol{v} \ll \boldsymbol{u}$. Hence there is also no such $\boldsymbol{v}$ in $M(\mathfrak{a}, \mathfrak{b})$. If $\boldsymbol{u}$ is a solution, it is therefore a minimal solution, and we put it into M^k. If $\boldsymbol{u}$ is not a solution, then we use it to generate L^{k+1}, which is the set of children of all these nonsolutions. If $d(\boldsymbol{u}) < 0$, the children $\boldsymbol{u}^i$ of $\boldsymbol{u}$ are generated by incrementing the component u_i on the left-hand side of $\boldsymbol{u}$, $i = 1 ,\ldots, m$. If $d(\boldsymbol{u}) > 0$, then the children $\boldsymbol{u}^j$ of $\boldsymbol{u}$ are generated by incrementing the component v_j on the right-hand side of $\boldsymbol{u}$, $j = 1 ,\ldots, n$. The algorithm stops, if L^k is empty. Since for all $\boldsymbol{u} \in L^k$ we have $s(\boldsymbol{u}) = k$, all sets L^k are disjoint.

```
FUNCTION FORT-HOM :

INPUT: A homogeneous equation HOM($\mathfrak{a}$, $\mathfrak{b}$), defined by $\mathfrak{a} = (a_1, \dots, a_m) \in \mathbf{N}^m$ and
       $\mathfrak{b} = (b_1, \dots, b_n) \in \mathbf{N}^n$.

  STEP 0 : $k$   := 1 ;
           $L^1$ := $\{\mathbf{u}^1, \dots, \mathbf{u}^{m+n}\} \subset \mathbf{N}_0^{m+n}$,
                    where every component of $\mathbf{u}^i$ is 0 but the i-th, which is 1;
           $L^2$ := $\emptyset$;
           $M^1$ := $\emptyset$;

  STEP 1 : FORALL $\mathbf{u} = (u_1, \dots, u_m, v_1, \dots, v_n) \in L^k$ DO:
           IF  $\neg(\exists s \in M^k : s \ll \mathbf{u})$   THEN
              IF  $\mathbf{u} \in S(\mathfrak{a}, \mathfrak{b})$
                THEN $M^k := M^k \cup \{\mathbf{u}\}$.
              ELSEIF  $d(\mathbf{u}) < 0$
                THEN $L^{k+1} := L^{k+1} \cup \{\mathbf{u}^i \mid \mathbf{u}^i = (u_1', \dots, u_m', v_1, \dots, v_n)$,
                                 $u_s' = u_s$ if $s \neq i$, $u_i' = u_i + 1$, $1 \le i \le m\}$;

              ELSEIF  $d(\mathbf{u}) > 0$
                THEN $L^{k+1} := L^{k+1} \cup \{\mathbf{u}^j \mid \mathbf{u}^j = (u_1, \dots, u_m, v_1', \dots, v_n')$,
                                 $v_s' = v_s$ if $s \neq j$, $v_j' = v_j + 1$; $1 \le j \le n\}$;

  STEP 2 : IF $L^{k+1} = \emptyset$  THEN  STOP ;
                                ELSE  $k$       := $k + 1$ ;
                                      $M^k$     := $M^{k-1}$ ;
                                      $L^{k+1}$ := $\emptyset$ ;
                                      GOTO STEP 1 .

OUTPUT : $M^k$ is the set of all minimal solutions to HOM($\mathfrak{a}$, $\mathfrak{b}$).

ENDOF FORT-HOM .
```

In the following paragraph we shall extend this algorithm to a system of inhomogeneous equations, defined by $\mathfrak{a}$, $\mathfrak{b}$ and $P \subset \mathbf{Z}$. Since we allow $P = \{0\}$, we may treat the homogeneous equation as a special case. There we shall also show that this algorithm can be significantly improved upon.

4.2 The Inhomogeneous Case

We now want to extend our version of Fortenbacher's algorithm to a system of inhomogeneous equations, as defined in 2.3. First we present the main idea which leads to a naive extension of FORT-HOM, whose correctness and termination are shown in Theorem 4-1 and Theorem 4-2, resp. Thereafter we discuss some inefficiencies and propose ways to avoid them. This leads to a second, more sophisticated algorithm, which inherits its correctness and termination from the first approach. It is shown to be complete in Lemma 4-5 and Theorem 4-6.

The main idea of the extension is to increment those proposals that satisfy the criterion of the homogeneous case, but to increment in a different manner. In the homogeneous case, a proposal $\mathbf{u}$ was only incremented, if there was no $\mathfrak{w} \in S(\mathfrak{a}, \mathfrak{b})$ with $\mathfrak{w} \ll \mathbf{u}$. In the following we shall call such a proposal **promising**. We may take over this criterion to the inhomogeneous case, because if there is such a $\mathfrak{w}$ and if any descendant s of $\mathbf{u}$ would be a solution to INHOM_p for some $p \in P$, then also $\mathfrak{w} \ll s$, and according to Theorem 2-8 s could not be a minimal solution. However, the incrementation-strategy changes: we increment the left-hand side of $\mathbf{u}$, if there is $p \in P$ with $d(\mathbf{u}) \le p$, increment the right-hand side, if there is $q \in P$ with $d(\mathbf{u}) \ge q$ and increment both sides, if there are $p, q \in P$ with $q < d(\mathbf{u}) < p$.

Hence we obtain a first extension of Fortenbacher's algorithm as:

```
FUNCTION NAIVE-FORT-INHOM :

INPUT :  A system of inhomogeneous equations with the same homogeneous
         parts, defined by $\mathfrak{a} = (a_1 , \dots , a_m) \in \mathbf{N}^m$,  $\mathfrak{b} = (b_1 , \dots , b_n) \in \mathbf{N}^n$
         and $P = \{p_1, \dots , p_r\} \subset \mathbf{Z}$ .

  STEP 0 :  $k$   := $1$ ;
            $P'$  := $P \cup \{0\}$ ;
            $L^1$ := $\{\mathfrak{v}^1, \dots , \mathfrak{v}^m , \mathfrak{w}^1, \dots , \mathfrak{w}^n\} \subset \mathbf{N}_0^{m+n}$
                  where every component of $\mathfrak{v}^i$ is 0, but the i-th, which is 1,
                  for $1 \le i \le m$,
                  and every component of $\mathfrak{w}^j$ is 0, but the (m+j)-th, which
                  is 1, for $1 \le j \le n$ .
            $L^2$              := $\emptyset$ ;
            ( $M_0^1$ = ) $M^1$  := $\emptyset$ ;
             $M_p^1$            := $\emptyset$  for all $p \in P$ ;
```

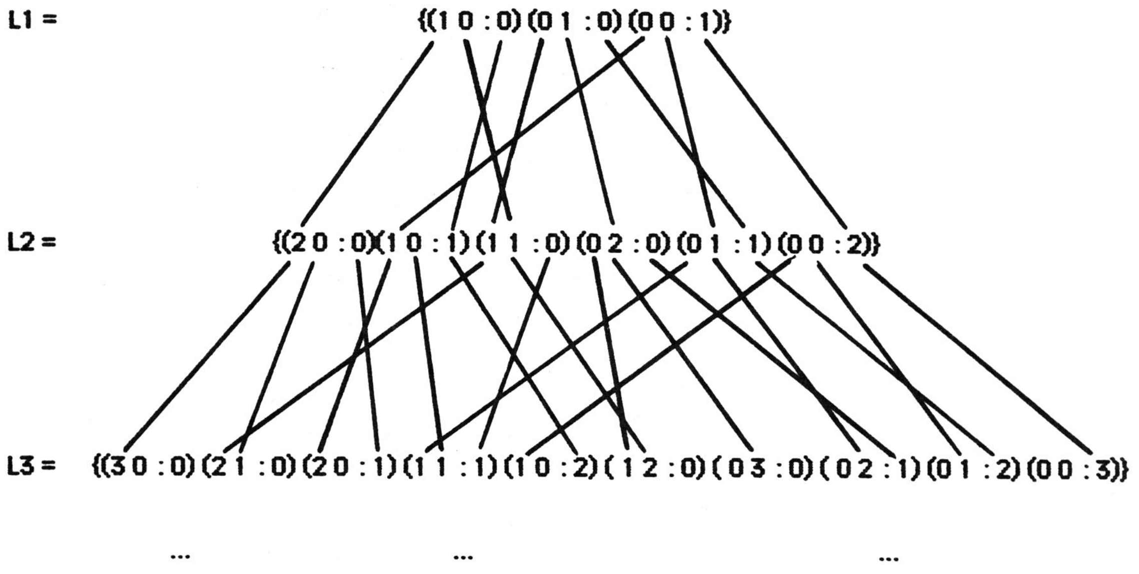
```
STEP 1 :  FORALL $\mathbf{u} = (u_1, \dots, u_m, v_1, \dots, v_n) \in L^k$ DO:
          IF   $\neg(\exists s \in M^k : s \ll \mathbf{u})$   THEN
            IF  $\mathbf{u} \in S(\mathfrak{a}, \mathfrak{b})$
              THEN $M^k := M^k \cup \{\mathbf{u}\}$.
            ELSE
              IF $\exists p \in P' : \mathbf{u} \in S_p(\mathfrak{a}, \mathfrak{b})$
                THEN  $M_p^k := M_p^k \cup \{\mathbf{u}\}$;
              IF        $\forall p \in P' : d(\mathbf{u}) \le p$
                THEN $L^{k+1} := L^{k+1} \cup \{\mathbf{u}^i \mid \mathbf{u}^i = (u_1', \dots, u_m', v_1, \dots, v_n)$, with
                                     $u_s' = u_i$ if $s \neq i$, $u_i' = u_i + 1$  for $1 \le i \le m\}$;
              ELSEIF   $\forall p \in P' : d(\mathbf{u}) \ge p$
                THEN $L^{k+1} := L^{k+1} \cup \{\mathbf{u}^j \mid \mathbf{u}^j = (u_1, \dots, u_m, v_1', \dots, v_n')$, with
                                     $v_s' = v_s$ if $s \neq j$, $v_j' = v_j + 1$  for $1 \le j \le n\}$
              ELSEIF   $\exists p, q \in P' : q \le d(\mathbf{u}) \le p$
                THEN $L^{k+1} := L^{k+1} \cup \{\mathbf{u}^i \mid \mathbf{u}^i = (u_1', \dots, u_m', v_1, \dots, v_n)$, with
                                     $u_s' = u_s$ if $s \neq i$,  $u_i' = u_i + 1$  for $1 \le i \le m\}$
                                   $\cup \{\mathbf{u}^j \mid \mathbf{u}^j = (u_1, \dots, u_m, v_1', \dots, v_n')$, with
                                     $v_s' = v_s$ if $s \neq j$,  $v_j' = v_j + 1$  for $1 \le j \le n\}$;

STEP 2 :  IF  $L^{k+1} = \emptyset$   THEN  STOP.
                              ELSE  $k$        $:= k + 1$;
                                    $M^k$      $:= M^{k-1}$
                                    $M_p^k$    $:= M_p^{k-1}$ for all $p \in P$;
                                    $L^{k+1}$  $:= \emptyset$ ;
                                    GOTO  STEP 1 .
```

OUTPUT: M^k is the set of all minimal solutions to HOM($\mathfrak{a}, \mathfrak{b}$) .
For all $p \in P$ is M_p^k the set of all minimal solutions to INHOM$_p$($\mathfrak{a}, \mathfrak{b}$).

ENDOF NAIVE-FORT-INHOM .

As an example consider the system of equations, defined by $\mathfrak{a} = (2, 2)$, $\mathfrak{b} = (3)$ and $P = \{-5, 5\}$, for which the algorithm generates the following proposals (the edges show, how each proposal is generated):



If L^k is the first empty set, i. e. the algorithm stops in this step, we define for the following proofs $L^s := \emptyset$ and $M_p^s := M_p^{k-1}$ for all $s \ge k$ and $p \in P \cup \{0\}$.

THEOREM 4-1: For all $p \in P \cup \{0\}$ and $k \ge 1$: every element of M_p^k is a correct and minimal solution to INHOM_p.

PROOF : Every element s of M_p^k is a solution to INHOM_p by construction. By Theorem 2-8 we know that s is a minimal solution, if there is no $v \in M(\mathfrak{a}, \mathfrak{b})$ with $v \ll s$. By construction of M_p^j for $j = s(s)$, there is no such v in M^j and since NAIVE-FORT-INHOM is just an extension of FORT-HOM, we know that $M^j = \{s \in M(\mathfrak{a}, \mathfrak{b}) | \; s(s) \le j\}$. However if $v \in M(\mathfrak{a}, \mathfrak{b})$ and $v \notin M^j$ then $s(v) > j$ and hence not $v \ll s$. Therefore s is a minimal solution to INHOM_p. ■

THEOREM 4-2: NAIVE-FORT-INHOM terminates after finitely many steps.

PROOF : The algorithm terminates, if no more proposals are generated. So we have to show, that every chain $\mathbf{u}^0 \ll \mathbf{u}^1 \ll \dots$, generated by the algorithm, has an upper bound, i. e. there is an $\mathbf{u}^i$ which is either a solution to the homogeneous equation $(d(\mathbf{u}^i) = 0)$ or no longer promising.

It is easy to show, that for every proposal $\mathbf{u}^i$, $d(\mathbf{u}^i)$ is bounded by

$$\min_p - \max_a < d(\mathbf{u}^i) < \max_p - \max_b.$$

Hence there are only finitely many values for $d(\mathbf{u}^i)$. So in every chain there is $\mathbf{u}^i$ with either $d(\mathbf{u}^i) = 0$ or there is $j < i$ with $d(\mathbf{u}^j) = d(\mathbf{u}^i)$. But in that case $\mathbf{u}^i - \mathbf{u}^j \ll \mathbf{u}^i$ is a solution to the homogeneous equation, and therefore $\mathbf{u}^i$ is not promising. ■

It is left to show that the algorithm is complete, i. e. every minimal solution to any of the equations is generated. However, we shall first discuss four ways to improve upon its efficiency and then show the completeness of a modified algorithm FORT-INHOM.

First, if an ancestor of a proposal **u** was a solution to INHOM_p for some $p \in P$, then neither **u** nor any descendant of **u** can be a minimal solution to this equation. Hence, when incrementing **u** it is enough to consider the set

$$P(\mathbf{u}) := \{p \in P \cup \{0\} \mid \text{no ancestor of } \mathbf{u} \text{ was a solution to } \mathrm{INHOM}_p\}$$

instead of P'.

Secondly, the following theorem tells us that every solution to the homogeneous equation, generated by the algorithm, is minimal. Hence we may test only those $\mathbf{u} \in L^k$ on minimality, that are no solutions.

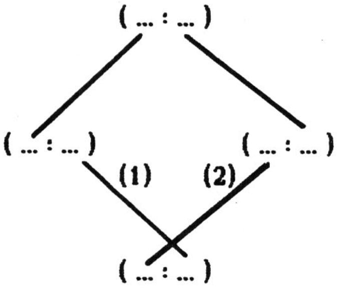
THEOREM 4-3: Any solution s to $\mathrm{HOM}(\mathfrak{a}, \mathfrak{b})$, generated by the algorithm, is minimal.

PROOF: Without loss of generality, $s = (c_1, ..., c_m, d_1, ..., d_n) \in M^k$ might have been generated from s' by incrementation of the first component. Since s' was incremented, it was not a solution to HOM and there was no $\mathfrak{v} \in M^{k-1}$ with $\mathfrak{v} \ll s'$, i. e. s' was promising. Now assume that there is a solution $\mathfrak{v} \ll s$ to HOM. Then the first component of $\mathfrak{v}$ is c_1, since otherwise $\mathfrak{v} \ll s'$ and s' would not be promising. Consider $\mathfrak{w} = s - \mathfrak{v}$, which is also a solution to HOM. It is $\mathfrak{w} \ll s$ and, since the first component of $\mathfrak{w}$ is $c_1 - c_1 = 0$, we have also $\mathfrak{w} \ll s'$, which contradicts that s' is promising. Hence there is no other solution $\mathfrak{v} \ll s$, and s is minimal in $S(\mathfrak{a}, \mathfrak{b})$. ■

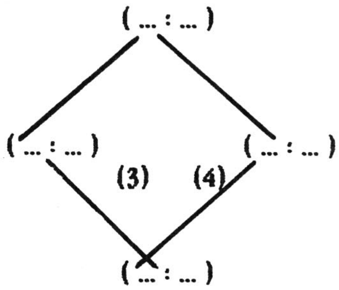
A third way to improve efficiency may be derived from the following observation: if a proposal $\mathbf{u}^i$ was generated by incrementing the i-th component of a proposal **u** and if $\mathbf{u}^i$ is either a solution to $\mathrm{HOM}(\mathfrak{a}, \mathfrak{b})$ or no longer promising, then we do not need to increment the i-th component of any other descendant of **u**, since that new proposal would not be promising, too. However, we did not find an easy and elegant way to integrate this idea with the three other improvements.

The last, but most important improvement of NAIVE-FORT-INHOM concerns the multiple generation of the same proposal. If there would be a unique history for each proposal, the total number of incrementations would be significantly reduced. Moreover, since we would not need to care about the distinctness of the proposals, we could use lists and list-append to implement the L^k and M_p^k, thus avoiding the computational overhead with sets and set-union. If we try to analyze, in what ways the same proposal might be generated, we find at least two schemata: Incrementation of two components on different sides (Fig. 1) and

incrementation of two components on the same side (Fig. 2).



(Fig. 1)



(Fig. 2)

In order to obtain a unique history for each proposal, we somehow have to prevent the confluences. In our modified algorithm, we do this by eliminating edges (1) and (3).

(1) is eliminated as follows: If a proposal **u** is to be incremented on both sides, than no descendant *w* of the right-hand side should ever be incremented on the left. We shall mark this by defining a predicate LEFT-INCREMENTABLE on the set of all proposals. (Note that edge (4) would be eliminated by exchanging left and right in this rule.)

Eliminating edge (3) yields proposal-histories that look as if both sides of a proposal were filled up from their back. (Analogous, eliminating edge (4) would fill both sides up from their front.) This is achieved by not incrementing all components on one side, but only those up to and including the first component different from 0. For each proposal **u** we shall mark the position of this first positive component on the left-hand side by $n_1(\mathbf{u})$ and on the right-hand side by $n_2(\mathbf{u})$. If there is no component different from 0 on one side, we use the position of the last component and define $n_1(\mathbf{u}) = m$ or $n_2(\mathbf{u}) = n$, resp.

With this incrementation-strategy, e. g. the left-hand side of (2 3 0 1 : ...) would be generated in the order (0 0 0 1 : ...)
(0 1 0 1 : ...)
(0 2 0 1 : ...)
(0 3 0 1 : ...)
(1 3 0 1 : ...)
(2 3 0 1 : ...).

Note that this process is likely to be dovetailed by filling the right-hand side up, depending on $P \cup \{0\}$.

By definition $n_1(\mathbf{u})$ and $n_2(\mathbf{u})$ denote the last component on the corresponding side, that is allowed to be incremented. It is interesting that this interpretation may be used to express LEFT-INCREMENTABLE(**u**) = false by setting $n_1(u) := 0$. This does not influence the following proofs.

To prove in Lemma 4-4 that the use of LEFT-INCREMENTABLE, n_1 and n_2 really

guarantees a unique way to generate each proposal, we need to describe their influence more formally: If $\mathbf{u} = (u_1, ..., u_m, v_1, ..., v_n)$ is an ancestor of a proposal $w = (w_1, ..., w_m, z_1, ..., z_n)$, then our rules obviously yield

(1) $u_i \leq w_i$ for $i = n_1(\mathbf{u})$

(2) $u_i = w_i$ for $n_1(\mathbf{u}) < i \leq m$

(3) $(\exists i : u_i < w_i) \Rightarrow$ LEFT-INCREMENTABLE($\mathbf{u}$)

(4) $v_j \leq z_j$ for $j = n_2(\mathbf{u})$

(5) $v_j = z_j$ for $n_2(\mathbf{u}) < j \leq n$.

We shall call proposals that satisfy these conditions for some w **w-incrementable**.

LEMMA 4-4: Let $w = (w_1, ..., w_m, z_1, ..., z_n)$ be a proposal. Then for all $k \geq 1$ and for all $\mathbf{u} \in L^k$ we have:

(a) If $\mathbf{u}$ is w-incrementable and $\mathbf{u}'$ is an ancestor of $\mathbf{u}$, then $\mathbf{u}'$ is also w-incrementable. (Transitivity)

(b) If $\mathbf{u}, \mathbf{x} \in L^k$ are w-incrementable and both are generated from the same proposal $\mathbf{u}' \in L^{k-1}$, then $\mathbf{u} = \mathbf{x}$.

(c) Moreover: If $\mathbf{u}, \mathbf{x} \in L^k$ are w-incrementable , then $\mathbf{u} = \mathbf{x}$.

(d) $\mathbf{u}$ is w-incrementable iff $\mathbf{u}$ is an ancestor of w.

PROOF: ad (a): If $\mathbf{u}'$ is an ancestor of $\mathbf{u}$, then $n_1(\mathbf{u}) \leq n_1(\mathbf{u}')$ and $n_2(\mathbf{u}) \leq n_2(\mathbf{u}')$. Therefore if $\mathbf{u}$ is w-incrementable, $\mathbf{u}'$ is also w-incrementable.

ad (b): Let $\mathbf{u} = (u_1, ..., u_m, v_1, ..., v_n) \in L^k$, generated from $\mathbf{u}' \in L^{k-1}$, be w-incrementable. Assume there is another child $\mathbf{x} = (x_1, ..., x_m, y_1, ..., y_n) \neq \mathbf{u}$ of $\mathbf{u}'$ and $\mathbf{x}$ is also w-incrementable. Then $\mathbf{u}$ and $\mathbf{x}$ differ in exactly two components. Each of these components is given either by n_1 (if it is on the left) or by n_2 (if it is on the right).

CASE 1: Both were generated by incrementation on the same side, w. l. o. g. on the left one. Since $\mathbf{u} \neq \mathbf{x}$, we may w. l. o. g. assume that $n_1(\mathbf{u}) < s := n_1(\mathbf{x})$. By construction of $\mathbf{u}$ and $\mathbf{x}$ we then have $u_s = x_s - 1$. Since both are w-incrementable, we get the contradiction $w_s = u_s < x_s \leq w_s$.

CASE 2: Both were generated by incrementation on different sides, w. l. o. g. $\mathbf{x}$ from the s-th component on the left and $\mathbf{u}$ from a component on the right side. Hence $u_s = x_s - 1$ and not LEFT-INCREMENTABLE($\mathbf{u}$). But since both are w-incrementable, this yields the contradiction $w_s = u_s < x_s \leq w_s$.

ad (c) : If $k = 1$, then $\mathbf{u}$ and $\mathbf{x}$ can be regarded as beeing generated from $\mathbf{0}$ and hence are equal by (b). If $k > 1$, then (a) yields that both were generated from proposals $\mathbf{u}'$ and $\mathbf{x}'$ that were $\mathbf{w}$-incrementable by (a). However by induction hypothesis we get that $\mathbf{u}' = \mathbf{x}'$, and hence by (b) $\mathbf{u}$ and $\mathbf{x}$ must be equal.

ad (d) : We already mentioned that $\mathbf{u}$ is $\mathbf{w}$-incrementable if $\mathbf{u}$ is an ancestor of $\mathbf{w}$. Now let $\mathbf{u} \in L^k$ be $\mathbf{w}$-incrementable. Since $\mathbf{u} \ll \mathbf{w}$, we have $k < s(\mathbf{w})$. Therefore $\mathbf{w}$ must have an ancestor in L^k which is also $\mathbf{w}$-incrementable. Now (c) yields that this ancestor is $\mathbf{u}$.

■

As a corollary of this lemma we have that each proposal $\mathbf{w}$ can be generated in exactly one way. Furthermore, since all starting-proposals are different, no multiple occurrences of the same proposal will appear, if we use lists and "append" instead of sets and set-union.

We now present our final algorithm that integrates the discussed improvements into NAIVE-FORT-INHOM. We will enclose lists in square-brackets and denote list-append by "$\sqcup$".

FUNCTION FORT-INHOM :

INPUT : A system of inhomogeneous equations with the same homogeneous parts, defined by $\mathfrak{a} = (a_1, \ldots, a_m) \in \mathbf{N}^m$, $\mathfrak{b} = (b_1, \ldots, b_n) \in \mathbf{N}^n$ and $P = [p_1, \ldots, p_r] \sqsubset \mathbf{Z}$.

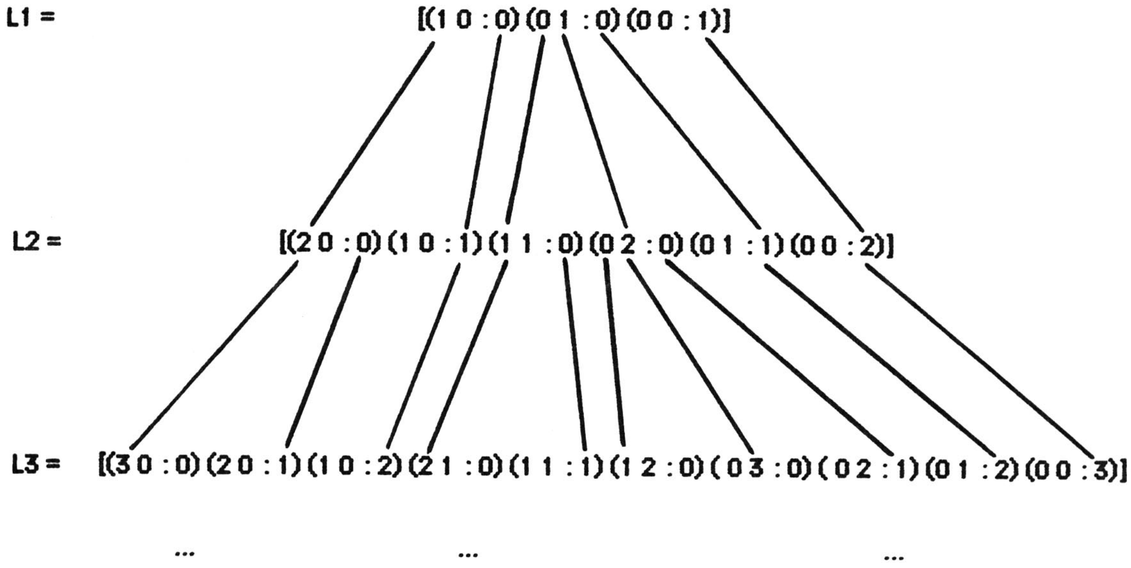
```
STEP 0 : $k$ := 1 ;
         $P'$ := $P \sqcup [0]$;
         $L^1$ := $[\mathbf{v}^1, \ldots, \mathbf{v}^m, \mathbf{w}^1, \ldots, \mathbf{w}^n] \sqsubset \mathbf{N}_0^{m+n}$
                 where every component of $\mathbf{v}^i$ is 0, but the i-th, which is 1,
                 $n_1(\mathbf{v}^i) = i$, $n_2(\mathbf{v}^i) = n$, LEFT-INCREMENTABLE($\mathbf{v}^i$),
                 $P(\mathbf{v}^i) = P'$, for $1 \le i \le m$ ,
                 and every component of $\mathbf{w}^j$ is 0, but the (m+j)-th, which
                 is 1, $n_1(\mathbf{w}^j) = m$, $n_2(\mathbf{w}^j) = j$, $\neg$ LEFT-INCREMENTABLE($\mathbf{w}^j$),
                 $P(\mathbf{w}^j) = P'$, for $1 \le j \le n$ .
         $L^2$ := nil ;
         ( $M_0^1$ = ) $M^1$ := nil ;
         $M_p^1$ := nil for all $p \in P$ ;
```

```
STEP 1 :  FORALL $u = (u_1, \dots, u_m, v_1, \dots, v_n) \in L^k$ DO:
          IF  $u \in S(a, b)$
             THEN $M^k := M^k \sqcup [u]$.
          ELSEIF  $\neg (\exists s \in M^k : s \ll u)$   THEN
             IF $\exists p \in P(u) : u \in S_p(a, b)$
               THEN  $M_p^k := M_p^k \sqcup [u]$;
                     $P(u) := P(u) \setminus [p]$;
             IF  $\forall p \in P(u) : d(u) < p$  AND LEFT-INCREMENTABLE($u$)
               THEN $L^{k+1} := L^{k+1} \sqcup [u^i \mid u^i = (u_1', \dots, u_m', v_1, \dots, v_n)$, with
                              $u_s' = u_i$ if $s \neq i$, $u_i' = u_i + 1$;  $P(u^i) = P(u)$,
                              LEFT-INCREMENTABLE($u^i$),
                              $n_1(u^i) = k$, $n_2(u^i) = n_2(u)$ for $1 \leq i \leq n_1(u)]$;
             ELSEIF  $\forall p \in P(u) : d(u) > p$
               THEN $L^{k+1} := L^{k+1} \sqcup [u^j \mid u^j = (u_1, \dots, u_m, v_1', \dots, v_n')$, with
                              $v_s' = v_s$ if $s \neq j$, $v_{jj}' = v_j + 1$,  $P(u^k) = P(u)$,
                              LEFT-INCREMENTABLE($u^j$) $\Leftrightarrow$
                                   LEFT-INCREMENTABLE($u$),
                              $n_1(u^j) = n_1(u)$, $n_2(u^j) = j$ for $1 \leq j \leq n_2(u)]$
             ELSEIF $\exists p,q \in P(u) : q < d(u) < p$ AND $\neg$ LEFT-INCREMENTABLE($u$)
               THEN $L^{k+1} := L^{k+1} \sqcup [u^j \mid u^j = (u_1, \dots, u_m, v_1', \dots, v_n')$, with
                              $v_s' = v_s$ if $s \neq j$, $v_j' = v_j + 1$,  $P(u^k) = P(u)$,
                              $\neg$ LEFT-INCREMENTABLE($u^j$),
                              $n_1(u^j) = n_1(u)$, $n_2(u^j) = j$ for $1 \leq j \leq n_2(u)]$
             ELSEIF  $\exists p,q \in P(u) : q < d(u) < p$ AND LEFT-INCREMENTABLE($u$)
               THEN $L^{k+1} := L^{k+1} \sqcup [u^i \mid u^i = (u_1', \dots, u_m', v_1, \dots, v_n)$, with
                              $u_s' = u_s$ if $s \neq i$,  $u_i' = u_i + 1$;  $P(u^i) = P(u)$,
                              LEFT-INCREMENTABLE($u^i$),
                              $n_1(u^i) = i$, $n_2(u^i) = n_2(u)$, for $1 \leq i \leq n_1(u)]$
                            $\sqcup [u^j \mid u^j = (u_1, \dots, u_m, v_1', \dots, v_n')$, with
                              $v_s' = v_s$ if $s \neq j$, $v_j' = v_j + 1$;  $P(u^j) = P(u)$,
                              $\neg$ LEFT-INCREMENTABLE($u^j$),
                              $n_1(u^j) = n_1(u)$, $n_2(u^j) = j$,  for $1 \leq j \leq n_2(u)]$;
STEP 2 :  IF  $L^{k+1}$ = nil THEN  STOP.
                         ELSE $k$        $:= k + 1$;
                              $M^k$      $:= M^{k-1}$
                              $M_p^k$    $:= M_p^{k-1}$  for all $p \in P$;
                              $L^{k+1}$  := nil ;
                              GOTO STEP 1 .
```

OUTPUT: M^k is the set of all minimal solutions to HOM($\mathfrak{a}$, $\mathfrak{b}$) .
For all $p \in P$ is $M_p{}^k$ the set of all minimal solutions to INHOM$_p$($\mathfrak{a}$, $\mathfrak{b}$).

ENDOF FORT-INHOM .

Applied to our example, this algorithm generates the following proposals:


L1 = [(1 0 : 0) (0 1 : 0) (0 0 : 1)]
L2 = [(2 0 : 0) (1 0 : 1) (1 1 : 0) (0 2 : 0) (0 1 : 1) (0 0 : 2)]
L3 = [(3 0 : 0) (2 0 : 1) (1 0 : 2) (2 1 : 0) (1 1 : 1) (1 2 : 0) (0 3 : 0) (0 2 : 1) (0 1 : 2) (0 0 : 3)]
...
...
...

Since the proposals generated by FORT-INHOM constitute a subset of those, generated by NAIVE-FORT-INHOM, correctness and termination of FORT-INHOM follow from Theorem 4-1 and Theorem 4-2, resp., if we prove that finally $M^k = M(\mathfrak{a}, \mathfrak{b})$. From Theorem 4-3 we obtain $M^k \subseteq M(\mathfrak{a}, \mathfrak{b})$. We shall now show with Lemma 4-5 and Theorem 4-6 that the algorithm generates all minimal solutions to any of the equations:

LEMMA 4-5: Let $\mathfrak{a}, \mathfrak{b}$ and P define a system of inhomogeneous equations and let be $p \in P \cup \{0\}$. Then for every minimal solution s to INHOM$_p$($\mathfrak{a}$, $\mathfrak{b}$) and for all $k \geq 1$, either $s \in M_p{}^k$ or there is $u \in L^k$, which is s -incrementable.

PROOF : Let $s = (c_1, \ldots, c_m, d_1, \ldots, d_n)$ be a minimal solution to INHOM$_p$($\mathfrak{a}$, $\mathfrak{b}$). Note that to show $s \in M_p{}^k$, it is enough to show $s \in L^k$, because by construction it is recognized as a solution and added to the solution-set $M_p{}^k$.
The proof is by induction on k in M^k:

Base step $k = 1$:
If there is $i \leq m$ with $c_i > 0$, then let t be the greatest index with this property and consider $\mathbf{u} := \boldsymbol{v}^t \in L^1$. Now either $\mathbf{u} = \mathrm{s}$ or $\mathbf{u}$ is s-incrementable.
However, if $c_i = 0$ for all $i \leq m$, there must be $j \leq n$ with $d_j > 0$, because $\mathrm{s} \neq \mathbf{0}$. Let t be the greatest index with this property and consider $\mathbf{u} := \boldsymbol{w}^t \in L^1$. Then either $\mathbf{u} = \mathrm{s}$ or $\mathbf{u}$ is s-incrementable.

Induction step:
If $\mathrm{s} \in M_p^{k-1}$ then also $\mathrm{s} \in M_p^k$.
If $\mathrm{s} \notin M_p^{k-1}$, then there is $\mathbf{u}' = (u_1', \dots, u_m', v_1', \dots, v_n') \in L^{k-1}$ which is s-incrementable. First we have to show that $\mathbf{u}'$ is incremented: Since s is minimal in M_p, Theorem 2-8 implies that s is also minimal in $M_0 \cup \{s\}$. Hence $\mathbf{u}' \ll \mathrm{s}$ yields that $\mathbf{u}' \notin M_0$ and that there is no $\boldsymbol{w} \in M^k \subseteq M(\mathbf{a}, \mathbf{b})$ with $\boldsymbol{w} \ll \mathbf{u}'$. Therefore $\mathbf{u}'$ will be incremented. Furthermore, $p \in P(\mathbf{u}')$ because s is minimal in M_p and $\mathbf{u}' \ll \mathrm{s}$, and therefore neither $\mathbf{u}'$ nor any ancestor of $\mathbf{u}'$ can be a solution to INHOM_p.
To choose $\mathbf{u} \in L^k$, generated from $\mathbf{u}'$ and s-incrementable, we have to consider four cases:

(1) $d(\mathbf{u}') < q$ for all $q \in P(\mathbf{u}')$, in particular $d(\mathbf{u}') < p$: Hence $\mathbf{u}'$ is only incremented on the left-hand side.

(2) $d(\mathbf{u}') > q$ for all $q \in P(\mathbf{u}')$, in particular $d(\mathbf{u}') > p$: Hence $\mathbf{u}'$ is only incremented on the right-hand side.

(3) There are $q_1, q_2 \in P(\mathbf{u}')$ with $q_1 < d(\mathbf{u}') < q_2$, but $\mathbf{u}'$ is not left-incrementable: Hence $\mathbf{u}'$ is only incremented on the right-hand side.

(4) There are $q_1, q_2 \in P(\mathbf{u}')$ with $q_1 < d(\mathbf{u}') < q_2$, but $\mathbf{u}'$ is left-incrementable: Hence $\mathbf{u}'$ will be incremented on both sides.

<u>CASE1:</u> There must exist $i \leq n_1(\mathbf{u}')$ with $u_i' < c_i$. Let t be the greatest index with this property. Then by incrementing this component a new proposal $\mathbf{u}$ will be generated, with $n_1(\mathbf{u}) = t \leq n_1(\mathbf{u}')$, $n_2(\mathbf{u}) = n_2(\mathbf{u}')$ and, because $\mathbf{u}' \ll \mathrm{s}$, either $\mathbf{u} = \mathrm{s}$ or $\mathbf{u}$ is s-incrementable.

<u>CASE 2:</u> Since there must exist $j \leq n_2(\mathbf{u}')$ with $v_j' < c_j$, we can choose $\mathbf{u}$ analogous to CASE 1: if t is the greatest index with $v_j' < d_j$ for $1 \leq j \leq n_2(\mathbf{u}')$, then $t \leq n_2(\mathbf{u}')$, and therefore by incrementing this component a new proposal $\mathbf{u}$ will be generated which is either equal to s or s-incrementable.

CASE 3: Since $\mathbf{u}'$ is s-incrementable but not left-incrementable, it must be $u_i' = c_i$ for $1 \leq i \leq m$ and hence $d(\mathbf{u}') > p$. Therefore we can choose $\mathbf{u}$ as in CASE 2, by incrementation on the right.

CASE 4: This case is more complicated. If $d(\mathbf{u}') > p$ and $u_i' = c_i$ for $1 \leq i \leq m$, we can choose $\mathbf{u}$ as in CASE 2 by incrementation on the right. But if $d(\mathbf{u}') > p$, and there exists i with $u_i < c_i$, then each proposal $\mathbf{u}^j$, generated by incrementation on the right-hand side of $\mathbf{u}'$ will not be left-incrementable and therefore not satisfy condition (3) of s-incrementability. However in this case or if $d(\mathbf{u}') < p$, we can choose $\mathbf{u}$ as in CASE 1, by incrementation on the left. In both subcases either $\mathbf{u} = \mathbf{s}$ or $\mathbf{u}$ is s-incrementable.

■

THEOREM 4-6: Function FORT-INHOM computes all minimal solutions to $\mathrm{INHOM}_p(\mathbf{a}, \mathbf{b})$ for every $p \in P$.

PROOF : Let $\mathbf{s}$ be a minimal solution to $\mathrm{INHOM}_p(\mathbf{a}, \mathbf{b})$. For $s(\mathbf{s})$, according to Lemma 4-3, either $\mathbf{s} \in M_p^k$ or there is $\mathbf{u} \in L^k$ with $\mathbf{u} \ll \mathbf{s}$. Since $\mathbf{u} \in L^k$ would imply $s(\mathbf{u}) = s(\mathbf{s})$, which contradicts $\mathbf{u} \ll \mathbf{s}$, it must be $\mathbf{s} \in M_p^k$.

■

5 : EXPERIMENTS WITH THE ALGORITHMS

5.1 General Remarks on Runtime-Measurements

Unfortunately, the SYMBOLICS 3600 does not provide a utility to measure pure CPU-time, i. e. runtime without the time needed for paging or garbage-collection. Hence we often measured very different runtimes for the same problem. A closer look showed that about every third or fourth result went astray. To overcome this problem and to get - at least to some extend - exact values, we chose the following method :

> Each task was measured eight times. From these values we only took the smaller ones up to the greatest value that did not differ more than 10 % from the next smaller one. The arithmetic mean of these values was taken for result.

Practice showed, that in most cases about six values out of eight were accepted and that these did not differ more than 5 % from their mean.

5.2 A Note to Problem-Formulation

As Huet mentioned in his paper, a different formulation of the initial problem might significantly improve upon runtime. He presented the following heuristics to get a fast execution :

-- $max_a \leq max_b$

-- $a_1 \geq a_2 \geq \ldots \geq a_m$

-- $b_1 \leq b_2 \leq \ldots \leq b_n$

Our experience validated these rules for the homogeneous as well as for the inhomogeneous case.

Practice with our version of Fortenbacher's algorithm showed, that in the average best results were achieved if

-- $a_1 \leq a_2 \leq \ldots \leq a_m$

-- $b_1 \leq b_2 \leq \ldots \leq b_n$

and for the inhomogeneous case, if in addition

-- $m \leq n$.

All values listed in the following are achieved with problem-formulations according to these rules; the time to get these formulations (e. g. sorting of the coefficients) is not included.

5.3 An Attempt of a Comparison

Analyzing the results of our various measurements, we observe the following points :

If many coefficients have no common divisors, so that the d_{ij} and e_{ij} get large, the algorithm of Huet is very slow, because many proposals (also nonminimal ones) must be generated.

The algorithm of Fortenbacher does not like large coefficients or large inhomogeneous parts, because they slow down convergence (see theorem 4-2).

5.4 Use for AC-Unification

The presented algorithms play a crucial role in AC-Unification. As pointed out in the introduction, the main difference between the unification-algorithms of Stickel [St 81] and Herold/Livesey/Siekmann [LS 76], [HS 85] is the different reduction to linear diophantine equations.

Besides the theoretical advantage (c. f. [Bü 85]), our findings show that the latter has important practical advantages :

-- The tables with the runtimes show that in almost every case it is much faster to compute all the solution-sets of the multiple inhomogeneous equations and the homogeneous one than the single solution-set of Stickel's large homogeneous equation. (This observation does not depend on our assumption that the inhomogeneous parts are not substantially larger than the homogeneous parts.)

-- The unifiers can be directly derived from the sets of minimal solutions, whereas in Stickel's algorithm a compatibility-computation has to be performed in addition.

ACKNOWLEDGEMENTS:

We would like to express our gratitude to Hans-Jürgen Bürckert, Jörg Siekmann and Manfred Schmidt-Schauß for a thorough reading of an earlier draft of this paper. Their constructive criticism has contributed much to the present form of this paper.

BIBLIOGRAPHY

[Bü 85] W. Büttner: "Unification in the Datastructure Multisets". Report, Siemens AG, Corporate Laboratories for Information Technology, München, 1985.

[CP 67] Clifford / Preston : "The Algebraic Theory of Semigroups", Vol 2. Providence, Rhode Island, 1967.

[Fa 84] F. Fages: "Associative-Commutative Unification". in Proc. of 7th CADE (ed. R.E. Shostak), pp. 194-208, 1984

[Fo 83] A. Fortenbacher: "Algebraische Unifikation" . Diplomarbeit, Universität Karlsruhe, 1983

[Go 1873] P. Gordon: "Über die Auflösung linearer Gleichungen mit reellen Coefficienten". in "Mathematische Annalen", Leipzig, 1873

[HS 85] A. Herold / J. Siekmann: "Unification in Abelian Semigroups". Universität Kaiserslautern, MEMO SEKI-85-III-KL, 1985

[Hu 78] G. Huet: "An Algortithm to generate the Basis of Solutions to Homogeneous Linear Diophantine Equations". in "Information Processing Letters", vol 7, no. 3, 1978

[La 85] D. Lankford: "A new Non-negative Integer Basis Algorithm For Linear Homogeneous Equations with Integer Coefficients". (unpublished)

[LS 76] M. Livesey / J. Siekmann: "Unification in Sets and Multisets". Universität Karlsruhe, MEMO SEKI-76-II, 1976

[Ma 78] K. Maiwand: "An Implementation of an AC-Unification Algorithm" Universität Karlsruhe, 1978

[St 75] M. Stickel: "A complete Unification Algorithm for Associative -Commutative Functions" . in Proc. 4th IJCAI, Tblisi, USSR, 1975

[St 81] M. Stickel: "A Unification Algorithm for Associative -Commutative Functions". in "Journal of the ACM" , vol 28, no 3, 1978

APPENDIX

Runtimes for various Homogeneous Equations

homogeneous		# of	Runtime of	Runtime of	FORT-HOM
coefficients		solutions	HUET-HOM [ms]	FORT-HOM [ms]	faster
(1)	(1 2)	2	4	3	•
(1)	(1 1 2)	3	5	4	•
(1 2)	(1 1 2)	7	9	9	
(3 7)	(5 8)	12	53	56	
(1 2)	(1 1 1 2 2)	13	15	16	
(1 2)	(1 1 2 10)	13	16	35	
(2 3 4)	(2 3 4)	13	24	34	
(5 7)	(3 8)	18	59	68	
(1 2)	(1 1 1 2 2 3)	19	29	27	•
(1 2)	(1 1 1 2 2 10)	19	26	54	
(1 2 3)	(4 5 6)	22	60	73	
(4 7)	(2 3 5)	22	95	96	
(1 1 2)	(1 1 2 2 3)	24	37	39	
(1 3 4)	(1 1 3 4)	27	65	66	
(1 1 2 2)	(1 1 1 2 2)	28	35	39	
(1 2 15)	(3 5 10)	29	113	123	
(1 2 5)	(1 2 3 4)	39	197	124	•
(1 1 1 2 5)	(1 2 3)	43	266	109	•
(1 1 2)	(1 2 10)	43	87	168	

(1 1 2 2)	(1 1 1 2 3)	44	101	83	•
(3 5 7)	(1 2 4)	44	306	176	•
(1 2 10)	(1 1 2 10)	50	153	236	
(2 5 9)	(3 7 8)	65	1 213	454	•
(2 5 9)	(1 2 3 7 8)	119	11 000	1 489	•
(2 2 2 3 3 3)	(2 2 2 3 3 3)	138	654	483	•
(2 5 9 10)	(3 7 8 10)	152	13 000	2 319	•
(1 2 2 5 9)	(1 2 3 7 8)	345	64 000	6 070	•
(1 2 10)	(1 1 1 2 2)	349	11 000	4 380	•

Runtimes for Systems of Inhomogeneous Equations

In columns seven and eight we list the runtimes of our two algorithms for various inhomogeneous problems. As mentioned before, such systems of inhomogeneous equations arise with the AC-Unification-algorithm of Herold and Siekmann [HS 85]. Instead, the AC-Unification-algorithm of Stickel [St 81] would have to deal with a larger homogeneous equation, which we solved with our versions of Huet's and Fortenbacher's algorithm. In column six we give the minimum of the two runtimes and mark in brackets, with which of the two algorithms it was achieved.

homogeneous		inhomogen.	number of solutions to			runtime in ms		
parts of the		parts of the	the hom.	each of the	Stickel's big	for Stickel's	of HUET	of FORT
equations		equations	equation	inhom. eqs	hom. equation	big hom. equ.	-INH	-INH
(1)	(2)	1	1	1 1	2	3 (F)	4	5
		1 2		1 1	3	4 (F)	5	7
		1 -1 -2		1 1 1	7	9 (H)	5	9
(1 2)	(1 1 2)	1 2	7	3 2	13	15 (H)	15	20
		1 2 3		3 2 4	19	27 (F)	18	36
		1 2 -1 -2		3 2 3 4	28	35 (H)	20	33
		-10		36	43	87 (H)	33	218
		10		6	13	16 (H)	28	156
		-10 10		36 6	50	153 (H)	59	283
		1 2 -10		3 2 36	349	4 380 (F)	44	247
		1 2 10		3 2 6	19	26 (H)	35	170
(2 5 9)	(3 7 8)	1 2	65	19 6	119	1 489 (F)	3 103	644
		1 2 -1 -2		19 6 24 39	345	6 070 (F)	5 048	835
		-10 10		32 15	152	2 319 (F)	4 155	1 447