REDUCIBILITY RELATIONSHIPS BETWEEN DECISION PROBLEMS FOR SYSTEM FUNCTIONS

by M. B. THURAIISINGHAM in Minneapolis, Minnesota (U.S.A.)

1. Introduction

In his investigation of the one-one equivalence between general combinatorial decision problems, CLEAVE [1] initiated the concept of system functions. They are defined on natural numbers and their values are finite sets of natural numbers. They have many properties in common with those arising from Gödel numbering various combinatorial systems such as semi-Thue-systems, Thue-systems, Turing-machines and Markov-algorithms. Thus the decision problems defined for these combinatorial systems can also be defined for system functions. Furthermore, if a property holds for a decision problem for all system functions, then it holds for that decision problem for all these combinatorial systems.

In [2], CLEAVE analysed the reducibility relationships between decision problems not by working on computing systems such as Turing-machines or semi-Thue-systems, but by examining the class of system functions defined by him.

Subsequently he showed that:

(i) If \( f \in S \) and \( n \in N \), then \( D[f](n) \leq D[f] \) (\( S \) is the class of all system functions, \( N \) is the set of all natural numbers, \( D[f](n) \) is the special confluence problem for \( f \) at \( n \), \( D[f] \) is the general derivability problem for \( f \)).

(ii) If \( f \in M \) and \( n \in N \), then \( D[f^{-1}](n) \leq C[f](n) \) (\( M \) is the class of all machine functions, where a machine function is a special kind of system function, \( C[f](n) \) is the special confluence problem for \( f \) at \( n \), \( D[f^{-1}](n) \) is the inverse special derivability problem for \( f \) at \( n \)).

From these results it can be seen that:

(i) There does not exist a system function such that the degrees of its special derivability problem and general derivability problem can be chosen independently.

(ii) There does not exist a machine function such that the degrees of its inverse special derivability problem and special confluence problem can be chosen independently.

These reducibility relationships between decision problems for system functions gives rise to the following question: Suppose the degrees of two decision problems for a system function cannot be chosen independently, then to what extent are they dependent on one another? Alternatively, if \( F[f] \) and \( G[f] \) are two decision problems for a system function \( f \) and \( F[f] \leq d G[f] \) for all \( f \in K \), where \( K \subset S \) and is any reducibility, then do the following hold:

(i) Given arbitrary recursively enumerable sets \( W_1 \) and \( W_2 \) such that \( W_1 \equiv d W_2 \), there is a \( g \in K \) such that \( W_1 \equiv d F[g] \) and \( W_2 \equiv d G[g] \)?

(ii) \( F[h] \leq d G[h] \) for some \( h \in K \)?

(iii) There is a stronger reducibility \( \equiv d_i \) such that \( F[f] \leq d_i G[f] \) for all \( f \in K \)?
In section 3 of this paper we consider the second result obtained by Cleave and give an answer to the above question by proving that:

(i) Given arbitrary recursively enumerable sets \( W_1 \) and \( W_2 \) such that \( W_1 \subseteq \_ \_ W_2 \), there exists an \( f \in M \) and \( n \in N \) such that \( W_1 \equiv_T D[f^{-1}] (n) \) and \( W_2 \equiv_T C[f] (n) \).

(ii) There is a \( g \in M \cap M^{-1} \) and \( n \in N \) such that \( D[g^{-1}] (n) \subseteq_T C[g] (n) \) and \( C[f] (n) \subseteq_T D[g^{-1}] (n) \), \( M \cap M^{-1} \) is the class of all inverse machine functions, where an inverse machine function is a special kind of machine function.

The essential points of our argument will be clearly exhibited in these proofs. The technique utilised in these proofs is a powerful tool for analysing the dependence relationships between decision problems for system functions. The preliminary definitions needed for these proofs will be given in Section 2. For the recursive function theory terminology stated in this paper we refer to [4].

2. Definitional preliminaries

The definition of system functions and the definitions in the theory of graphs given in this section have been obtained from [1] and [5]. In section 3 these graph theoretic concepts will be employed in formulating certain algorithms.

Let \( f : N \to P_\omega(N) \), where \( N \) is the set of all natural numbers and \( P_\omega(N) \) is the set of all finite subsets of \( N \). For \( X \in P_\omega(N) \) define \( f(X) = \bigcup \{ f(x) : x \in X \} \). For each \( x \in N \) define \( f^0(x) = x, \) \( f^{k+1}(x) = f(f^k(x)) \); \( f^{-1} \), the inverse of \( f \), is defined by \( f^{-1}(x) = \{ y : x \in f(y) \} \). By \( y \) is directly derivable by \( f \) from \( x \) we mean \( y \in f(x) \). By \( y \) is derivable by \( f \) from \( x \) (denoted by \( y \in C_f x \) or \( x \in C_f y \)) we mean that either \( y = x \) or \( y \in f(x) \) or there exist \( y_1, y_2, \ldots, y_n \) \( (n \geq 1) \) such that \( y_1 \in f(x), y \in f(y_n) \) and for each \( i \) \( (1 \leq i \leq n - 1), y_{i+1} \in f(y_i) \).

A system function is a function \( f : N \to P_\omega(N) \) such that there exist recursive functions \( a \) and \( b \) such that for all \( x, f(x) = D_{a(x)} \) and \( f^{-1}(x) = D_{b(x)} \), where \( D_n \) is the \( n^{\text{th}} \) finite set in some standard enumeration.

A system function \( f \) such that \( f(x) \) for each \( x \) has almost one member is called a machine function. Clearly system functions that arise from deterministic combinatorial systems such as Turing-machines or Markov-algorithms are machine functions. A machine function \( f \) such that \( f^{-1} \) is also a machine function is called an inverse machine function. The class of all system functions (machine functions, inverse machine functions, respectively) will be denoted by \( S \) \( (M, M \cap M^{-1}, \text{respectively}) \).

Some of the decision problems for a system function \( f \) are:

(i) \( D[f] = \{ (x, y) : y \in C_f x \} \) (general derivability problem for \( f \)),

(ii) \( D[f] (n) = \{ x : x \in C_f n \} \), where \( n \in N \) (special derivability problem for \( f \) at \( n \)),

(iii) \( C[f] (n) = \{ x : (\exists y) (y \in C_f x \land y \in C_f n) \} \), where \( n \in N \) (special confluence problem for \( f \) at \( n \)),

(iv) \( D[f^{-1}] (n) = \{ x : n \in C_f x \} \), where \( n \in N \) (inverse special derivability problem for \( f \) at \( n \)).

Let \( D \) be a digraph whose points are in \( N \) and whose lines are ordered pairs of natural numbers. By \( x \in D \) we mean that \( x \) is a point in \( D \). If \( x \in D \) and \( y \in D \), then by
x → y (D) we mean that either x = y or (x, y) is a directed line in D or there exist distinct points y₁, y₂, ..., yₙ (n ≥ 1) such that (x, y₁), (y₁, y₂), ..., (yₙ₋₁, yₙ) and (yₙ, y) are all directed lines in D. If x ∈ D, then D(x) is the component of D containing x. By x ⪯ y we mean that x and y belong to different components of D. The in-degree (out-degree, respectively) of a point x is the number of points y such that (y, x) ((x, y), respectively) is a directed line in D. A point x is a root (leaf, respectively) of D if its in-degree (out-degree, respectively) is 0. r(x) (t(x), respectively) is the least number y such that y is a root (leaf, respectively) of D and y → x (D) (x → y (D), respectively).

A digraph is labelled if some of its points are distinguished from one another by names drawn from some given infinite list. By the term introduce labels L₁, L₂, ..., Lₙ (n ≥ 1) to D is meant the following: Find the least n numbers x₁ < x₂ < ... < xₙ which are not points in D, adjoin these numbers as new points so that each such point forms a new component, and name x₁ by L₁, x₂ by L₂, ..., xₙ by Lₙ.

A line graph is a digraph whose points have in-degree and out-degree of atmost 1. By the term extend the line graph L to the line graph L₁ is meant the following: Let t₁, t₂, ..., tₖ be the leaves of L and r₁, r₂, ..., rₘ be its roots; find the least k + m numbers x₁ < x₂ < ... < xₖ₊ₘ which are not points in L; adjoin these numbers as new points and join the lines (t₁, x₁), (t₂, x₂), ..., (tₖ, xₖ), (xₖ₊₁, r₁), (xₖ₊₂, r₂), ..., (xₖ₊ₘ, rₘ); the resulting graph is L₁ (see Figure 1).

3. Dependence relationships between decision problems for system functions

In this section we will prove the following

Result (x). (i) Given arbitrary recursively enumerable sets Wₑ₁ and Wₑ₂, where Wₑ₁ ≤_T Wₑ₂. Then there exists an f ∈ M such that Wₑ₁ ≡_T D[f] (0) and Wₑ₂ ≡_T C[f] (0) (Wₑ is the eth recursively enumerable set in some standard enumeration).

(ii) There exists g ∈ M ∩ M⁻¹ such that D[g⁻¹] (0) ≤_T C[g] (0) and C[g] (0) ≤_T D[g⁻¹] (0).
Proof. (i) We construct functions \( f_1 : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{P}(\mathbb{N} \times \mathbb{N}) \) and \( f_2 : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{P}(\mathbb{N} \times \mathbb{N}) \) as follows (the graphical representation of the construction of \( f_1 \) is shown in figure 2):

\[
\begin{align*}
\forall (x, y) \in \mathbb{N} \times \mathbb{N}, \quad f_1(x, y) = & \left\{ \begin{array}{ll}
(x, y + 1), & \text{if } x \neq 0 \land \neg T(e_1, x, y) \\
(0, x + y), & \text{if } x \neq 0 \land T(e_1, x, y) \\
\emptyset, & \text{if } x = 0 \land y = 0 \\
(x, y - 1), & \text{if } x = 0 \land y \neq 0.
\end{array} \right.
\end{align*}
\]

The construction of \( f_2 \) is similar to the construction of \( f_1 \) with \( e_1 \) in place of \( e_1 \).

![Diagram](image.png)

Note that \( W_e = \{ x : (3y) T(e, x, y) \} \), where \( T \) is the Kleene's \( T \)-predicate.

Define a function \( f : \mathbb{N} \rightarrow \mathcal{P}(\mathbb{N}) \) as follows:

\[
\begin{align*}
f(x) = & \left\{ \begin{array}{ll}
2y : y \in f_1(x/2), & \text{if } x \text{ is divisible by } 2, \\
2y + 1 : y \in f_2((x - 1)/2), & \text{if } x - 1 \text{ is divisible by } 2.
\end{array} \right.
\end{align*}
\]

Note that by \( y \in f_1(x) \) (or \( f_2(x) \)) we mean that \( (u_1, v_1) \in f_1(u_2, v_2) \) (or \( f_2(u_2, v_2) \)), where \( y = \tau(u_1, v_1), x = \tau(u_2, v_2) \), and \( \tau \) is a 1-1 recursive function from \( \mathbb{N}^2 \) onto \( \mathbb{N} \).

It can be easily seen that \( f_1, f_2, f \) all belong to \( M \). Furthermore as \( W_{e_1} \preceq_T W_{e_2} \), from the constructions of these functions we can easily show that

(i) \( D[f_1^{-1}] (0) \equiv_T W_{e_1} \),
(ii) \( C[f_1^{-1}] (0) \equiv_T W_{e_2} \),
(iii) \( D[f_2^{-1}] (0) \equiv_T D[f_1^{-1}] (0) \),
(iv) \( C[f_2^{-1}] (0) \equiv_T C[f_1^{-1}] (0) \).

From these results we have that \( D[f^{-1}] (0) \equiv_T W_{e_1} \) and \( C[f] (0) \equiv_T W_{e_2} \).

(ii) In this proof a \( g \in M \cap M^{-1} \) will be constructed such that if \( G_1 = D[g^{-1}] (0) \) and \( G_2 = C[g] (0) \), then \( G_1 \preceq_T G_2 \) and \( G_2 \preceq_T G_1 \). The proof will be divided into two parts. The first part will consist of a programme in which labelled line graphs \( B^0, B^1, B^2, \ldots \) and sets \( G^n_1, G^n_2, G^2_1, \ldots \) \((i \in \{1, 2\})\) will be constructed with the following properties:

(i) \( G^n_1 \preceq_T G^n_2 \) and \( G^n_2 \preceq_T G^n_1 \),
(ii) There is a recursive function \( \alpha \) such that for each \( n \), \( \alpha(n) \) is the Gödel number of \( B^n \).
(iii) $B^{n+1}$ is an extension of $B^n$, i.e. all points of $B^n$ are points of $B^{n+1}$ and if $x, y$ are points of $B^n$, then there is a line from $x$ to $y$ in $B^m$ ($m > n$) if and only if there is a line from $x$ to $y$ in $B^{n+1}$. Furthermore $B^{n+1}$ contains as a point the least number which is not a point of $B^n$.

(iv) If $B = \bigcup \{B^n: n \in \mathbb{N}\}$, where

$$B^i \cup B^j = \begin{cases} B^j & \text{if } i < j, \\ B^i & \text{if } i \geq j, \end{cases}$$

then for any point $x$ of $B^n$, all lines incident with $x$ in $B$ are lines of $B^{n+1}$.

(v) The function $g \in M \cap M^{-1}$ is defined as follows:

$$g(x) = \{y: x \rightarrow y (B)\} = \{y: x \rightarrow y (B^{x+1})\}$$

and similarly $g^{-1}(x) = \{y: y \rightarrow x (B)\} = \{y: y \rightarrow x (B^{x+1})\}$.

(vi) The sets $G^n_1$ and $G^n_2$ are defined by

$$G^n_1 = \{x: x \rightarrow 0 (B^n)\}, \quad G^n_2 = \{x: x \rightarrow 0 (B^n) \lor 0 \rightarrow x (B^n)\}.$$  

For $i \in \{1, 2\}$, define $G_i = \bigcup \{G^n_i: n \in \mathbb{N}\}$, where

$$G^i_j \cup G^k_j = \begin{cases} G^i_j & \text{if } j > k, \\ G^k_j & \text{if } k \geq j. \end{cases}$$

(vii) If at any instant the line graph constructed up to that stage is $B^m$, then the subset of $G_i$ ($1 \leq i \leq 2$) constructed up to that stage will be denoted by $G^m_i$.

(viii) Labels will be drawn from the following infinite sets of markers: $\{P^e; e \geq 0\}$, $\{P^e_2; e \geq 0\}$. The label $P^e_1$ will be used to show that $G_1 \equiv_T G_2$. The label $P^e_2$ will be used to show that $G_2 \equiv_T G_1$. The dependence of a label on another label will be defined by induction.

The second part of the proof consists of three lemmas by means of which it will be shown that

(i) For no $e$ is it true that $q_e$ is a truth table reduction function for the truth table reduction of $G_1$ to $G_2$, where $q_e$ is the $e^{th}$ partial recursive function of one variable in some standard enumeration.

(ii) For each $e$ the following condition (θ) holds:

$$\theta \quad (\exists y) (y \in G_2 \equiv y \in W^e_{G_1}),$$

where $W^e_{G_1}$ is the $e^{th}$ recursively enumerable set relative to $G_1$ in some standard enumeration.

Clearly, if (i) and (ii) hold, then $G_1 \equiv_T G_2$ and $G_2 \equiv_T G_1$.

We will only state these lemmas here. Their proofs can be obtained in [5]. These proofs use an argument similar to the priority argument by Friedberg [3].

Programme.

Stage 0. $B^0$ consists of the points 0 and 1 with 1 labelled $P^0_1$.

Stage 1. Introduce the label $P^0_2$ to $B^0$. Note that no dependence relations between labels are defined in stages 0 and 1.

Stage $2n$ ($n \geq 1$).

Step 1. Introduce the label $P^n_1$ to $B^{2n-1}$ and extend the resulting graph to $B^{2n-1}$. 

Step 2. Find the least number $e \leq n$ such that there are $x, u^1, u^2, \ldots, u^m, z$, all \( \leq n \), satisfying $A(z, e, x, u^1, u^2, \ldots, u^m, n)$, where $A$ is the conjunction of the following conditions $A_1$, $A_2$ and $A_3$:

$A_1 \equiv x$ is labelled $P_1$ in $B^{2^n-1}$,

$A_2 \equiv$ there exists an $m$-ary Boolean function $\Psi$ such that $\langle u^1, u^2, \ldots, u^m; \Psi \rangle$ is the tt-condition $U(z)$, where if $z$ is the Gödel number of a Turing-machine computation, then $U(z)$ is the output of that Turing-machine.

If there is no such an $e$, then set $B^{2^n} = B^{2^n-1}$. If there is such an $e$, define

$e_n = (ue) (\exists z, x, u^1, u^2, \ldots, u^m, \text{ all } \leq n) A(z, e, x, u^1, u^2, \ldots, u^m, n),

x_n = (ux) (\exists y, u^1, u^2, \ldots, u^m, \text{ all } \leq n) A(z, e, x, u^1, u^2, \ldots, u^m, n),

u^1_n = (u u^1) (\exists z, u^2, \ldots, u^m, \text{ all } \leq n) A(z, e, x, u_1^1, u^2, \ldots, u^m, n),

\ldots

u^m_n = (u u^m) (\exists z, u^2, \ldots, u^m, \text{ all } \leq n) A(z, e, x, u_1^m, u^2, \ldots, u^m, n).

For convenience, let $e_n, x_n, u^1_n, u^2_n, \ldots, u^m_n$ be $e, x, u^1, u^2, \ldots, u^m$, respectively. Then there is an $m$-ary Boolean function $\Psi$ such that $q_d(x)$ is the tt-condition $\langle u^1, u^2, \ldots, u^m; \Psi \rangle$. In step 3 it will be ensured that

$x \in G_2^{2^n}$ is the tt-condition $\langle u^1, u^2, \ldots, u^m; \Psi \rangle$ is satisfied by $G_2^{2^n}$.

The application of step 3 to $e$ will be called an attack on $P_1^{\ast}$.

Step 3.

(a) Delete $P_1^{\ast}$.

(b) Re-introduce all $P_i^{\ast}$ such that $P_i^{\ast}$ depends on $P_1^{\ast}$ (i.e. if such labels are already assigned to points, delete them and re-introduce them; if not, introduce them). Let the resulting graph be $B^+$.

(c) Construct $B^{2^n}$ from $B^+$ as follows: For each $i$ ($1 \leq i \leq m$) let

$b_i = \begin{cases} 1 & \text{if } u^i \in G^{2^n-1}, \\ 0 & \text{otherwise}. \end{cases}$

Compute $\Psi(b_1, b_2, \ldots, b_m)$.

Case 1. $\Psi(b_1, b_2, \ldots, b_m) = 1$. If there is a $u^i$ ($1 \leq i \leq m$) such that $B^+(u^i)$ is labelled $P_1^{\ast}$ and $B^+(u^i) \cap B^+(0) = 0$, then execute the following programme (d). If not, set $B^{2^n} = B^+$.

Programme (d).

1. Let $u^{i_1}, u^{i_2}, \ldots, u^{i_t}$ ($1 \leq i_1, i_2, \ldots, i_t \leq m$) be all such $u^i$. Let $C = \{u^{i_1}, u^{i_2}, \ldots, u^{i_t}\}$. Go to 2.

2. If $C = 0$, then the resulting graph is $B^{2^n}$. If not go to 3.

3. Let $u^{i(k)}$ ($1 \leq k \leq t$) be the least element of $C$. Let $B^{\ast}(u^{i(k)})$ be labelled $P_1^{\ast}$.

(i) If $q > e$ or $(q = e$ and $r = 2)$, then delete $P_1^{\ast}$ and re-introduce it.

(ii) If $q < e$, then record that $P_1^{\ast}$ depends on $P_1^{\ast}$.

Set $C = C \setminus \{u^{i(k)}\}$ (i.e. the new value of $C$ is $C \setminus \{u^{i(k)}\}$, where $C$ is the old value of $C$). Go to 2.
REDUCIBILITY RELATIONSHIPS BETWEEN DECISION PROBLEMS FOR SYSTEM FUNCTIONS

Case 2. \( \mathcal{P}(b_1, b_2, \ldots, b_m) = 0 \). If there is a \( u^i \) \((1 \leq i \leq m)\) such that either \( B^+(u^i) \) is labelled \( P_i \) and \( B^+(u^i) \cap B^+(0) = 0 \) or \( u^i \in B^+(x) \), then execute the programme \((\gamma)\). If not, set \( B^{2n} = B^+ \cup (t(x), r(0)) \) (i.e. \( B^{2n} \) results from \( B^+ \) by adjoining the line \( (t(x), r(0)) \)).

Programme \((\gamma)\).
1. If there is a \( u^i \) \((1 \leq i \leq m)\) such that \( B^+(u^i) \) is labelled \( P_i \) and \( B^+(u^i) \cap B^+(0) = 0 \), then go to 2. If not set \( B^* = B^+ \) and go to 5.
2. Let \( u^{i_1}, u^{i_2}, \ldots, u^{i_s} \) \((1 \leq i_1, i_2, \ldots, i_s \leq m)\) be all such \( u^i \), let \( C = \{ u^{i_1}, u^{i_2}, \ldots, u^{i_s} \} \). Go to 3.
3. If \( C = \emptyset \), set the resulting graph to be \( B^* \) and go to 5. If not, go to 4.
4. Let \( u^k \) \((1 \leq k \leq t)\) be the least element of \( C \). Let \( B^*(u^k) \) be labelled \( P_i \). If \( q > e \) or \( q = e \) and \( r = 2 \), then delete \( P_i \) and re-introduce it.
   (i) If \( q < e \), then record that \( P_i \) depends on \( P_k \). Set \( C := C \setminus \{ u^k \} \). Go to 3.
5. If there is a \( u^j \) \((1 \leq j \leq m)\) such that \( u^j \in B^*(x) \), then go to 6. If not, set \( B^{2n} = B^* \cup (t(x), r(0)) \).

Step 2
1. Introduce the label \( P_2^* \) to \( B^{2n} \) and extend the resulting graph to \( B^{2n} \).
2. Find the least number \( e \leq n \) such that there are \( x, z \) both \( \leq n \) satisfying \( A(z, e, x, n) \), where \( A \) is the conjunction of the following conditions \( A_1 \) and \( A_2 \).
   \[ A_1 \equiv T^G_1, n(e, x, z), \]
   \[ A_2 \equiv x \text{ is labelled } P_i \text{ in } B^{2n} \text{ and } B^{2n}(x) \cap B^{2n}(0) = 0. \]
If there is no such an \( e \), set \( B^{2n+1} = B^{2n} \).
If there is such an \( e \), define \( e_n, x_n \) as in stage \( 2n \). For convenience let \( e_n, x_n \) be \( e, x \) respectively. In step 3 it will be ensured that \( x \in G_{2n}^2 \equiv x \in W_{2n}^G \).
The application of Step 3 to \( e \) will be called an attack on \( P_2^* \).

Step 3
(a) Re-introduce all labels \( P_i^* \) such that \( P_i^* \) depends on \( P_i \).
(b) Join the line \((t(0), r(x))\). Let the resulting graph be \( B^+ \).
(c) Construct \( B^{2n+1} \) from \( B^+ \) as follows: During the computation of \( T^G_i, n(e, x, z) \) (see Step 2), answers to questions of the form \( \text{"Does } u \in G_{2n}^2 \text{?} \) would have been needed for certain \( u < z \). Let \( u_1, u_2, \ldots, u_s \) be all such \( u \) such that for each \( i \) \((1 \leq i \leq s)\), \( u_i \in G_{2n}^2 \). If there is a \( u_i \) \((1 \leq i \leq s)\) such that \( B^+(u_i) \) is labelled \( P_i^* \) and \( P_i^* \neq P_2^* \) then execute the programme \((\gamma)\). If not set \( B^{2n+1} = B^+ \).
Programme (q).

1. Let $u_{i_1}, u_{i_2}, \ldots, u_{i_r} (1 \leq i_1, i_2, \ldots, i_r \leq s)$ be all such $u_i$. Let $C = \{u_{i_1}, u_{i_2}, \ldots, u_{i_r}\}$. Go to 2.

2. If $C = \emptyset$, then the resulting graph is $B^{2n+1}$. If not, go to 3.

3. Let $u_{i_k} (1 \leq k \leq t)$ be the least element of $C$. Let $B^+(u_{i_k})$ be labelled $P^q$. 
   (i) If $q > e$, then delete $P^q$ and re-introduce it.
   (ii) If $q < e$ or $(q = e$ and $r = 1)$, then record that $P^q$ depends on $P^2$.
   Set $C = C \setminus \{u_{i_k}\}$. Go to 2.

This ends the construction of stage $2n + 1$ and hence the programme.

Definition. A label $L$ is fixed at a stage numbered $H$ if (i) it is assigned to the same point at all stages numbered $n \geq H$ or (ii) it remains unassigned at all stages numbered $n \geq H$.

Lemma 1.
(i) For each $e$ there is a stage $H^*_1$ at which all labels $P^1_i (i \leq e)$ are fixed.
(ii) For each $e$ there is a stage $H^*_2$ at which all labels $P^2_i (i \leq e)$ are fixed and assigned to points.

Lemma 2.
(i) For each $e$, if $q_e$ is total, then $P^1_e$ is fixed and unassigned at stage $H^*_1$.
(ii) For each $e$ such that $q_e$ is total there is an $x$ such that
   
   \[ x \in G_1 \equiv \text{the tt-condition } q_e(x) \text{ is satisfied by } G_2 \]
   
   (i.e. for no $e$ is it true that $q_e$ is the tt-reduction function of the tt-reduction of $G_1$ to $G_2$).

Lemma 3. For each $e$ there is an $x$ such that
   
   \[ x \in G_2 \equiv x \in W^{G_1}_{e_1} \]
   
   (i.e. $(\forall e) (\bar{G}_2 \neq W_{e_1}^{G_1})$).

The lemmas 2 and 3 imply that $G_1 \leq_{tt} G_2$ and $G_2 \leq_{tt} G_1$.

References


M. B. Thuraisingham
P.O. Box 21064
Columbia Heights
Minneapolis, Minnesota 5542
U.S.A.

(Eingegangen am 12. März 1986)