An Abstract Model for Parallel Computations: Gandy's Thesis*

to appear in: The Monist
Special Issue on Philosophy of Computer Science
edited by G. Longo

Wilfried Sieg and John Byrnes
sieg+@cmu.edu and byrnes+@cmu.edu
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA 15213
USA

* Dedicated to the memory of Robin Gandy. – The initial steps towards this presentation were taken ten years ago; cf. [Sieg 1990] and [Byrnes 1993]. They were discussed extensively with Robin Gandy; John Shepherdson made some helpful remarks at that stage. The more radical simplifications emerged during the last six months.

0. Introduction. In his classical paper *On computable numbers (with an application to the Entscheidungsproblem)* **Turing analyzed what can be done** by a human computor in a routine, "mechanical" way. He argued that mechanical operations obey *locality conditions* and are carried out on configurations satisfying *boundedness conditions.* Processes meeting these restrictive conditions can be shown to be computable by a Turing machine.[1] Turing viewed memory limitations of computors as the ultimate reason for the restrictive conditions. In contrast, Gandy analyzed in his paper *Church's* ***Thesis and principles for mechanisms*** **what can be done by "discrete** deterministic mechanical devices" and appealed to physical considerations to motivate restrictive principles. These devices include crucially ones that carry out parallel processes, e.g., cellular automata. Following Turing's methodological ways and guided by Conway's "Game of life" as a paradigm, Gandy formulated four restrictive principles and proved that devices satisfying them are computationally equivalent to Turing machines.

The definitional preliminaries in Gandy's paper are rather lengthy; Shepherdson wrote in 1988: "Although Gandy's principles were obtained by a very natural analysis of Turing's argument they turned out to be rather complicated, involving many subsidiary definitions in their statement. In following Gandy's argument, however, one is led to the conclusion that that is in the nature of the situation." The "nature of the situation" is actually not as complex. The presentation can be simplified by choosing definitions appropriately, focusing sharply on the central informal ideas, and using one key suggestion made by Gandy in an Addendum to his paper; see Note 1. Our simplifications do not change the "form of presentation", i.e., the states of mechanical devices are given by hereditarily finite sets as in Gandy's paper. However, the subsidiary definitions are streamlined significantly, and of the four principles Gandy used only the principle of local causality is explicitly retained to characterize "Gandy Machines". It is formulated in two separate parts, namely, as the principle of *Local Causation* and that of *Unique* ***Assembly .***

Our presentation of Gandy machines is not given solely for Gandy machines' sake; rather, we want to address the broader issue of obtaining a

---

[1] **For details of this analysis cf. [Sieg, 1994], [Sieg & Byrnes, 1998], but also note 5 below. Roughly speaking, the first conditions make operations *local* in the sense that the "next "operation must take place within a fixed finite neighborhood; the second conditions require that the configurations (on which the operations are carried out) have a fixed *bounded* size. Turing machines obviously satisfy these conditions.**

truely abstract model for parallel computations. In [Lamport & Lynch] one finds this observation:

The theory of sequential computing rests upon fundamental concepts of computability that are independent of any particular computational model. If there are any such fundamental formal concepts underlying distributed computing, they have yet to be developed.... Nevertheless, one can make some informal observations that seem to be important.

Underlying almost all models of concurrent systems is the assumption that an execution consists of a set of discrete events, each affecting only part of the system's state. Events are grouped into processes, each process being a more or less completely sequenced set of events sharing some common locality in terms of what part of the state they affect. For a collection of autonomous processes to act as a coherent system, the processes must be synchronized, (p. 1166)

It seems to us that Gandy's analysis is conceptually convincing and provides a sharp mathematical form of the informal assumption(s) "underlying almost all models of concurrent systems". Indeed, the central informal idea is just this: operations on states are composed of local transformations that modify "parts" of bounded size, and the resulting configurations are then uniquely assembled into the next state of the system. Turing and Gandy both focus on mechanical processes operating deterministically on discrete states; whereas Turing arrived at his distinctive analysis of human calculability by exploiting broad limitations of the computing agent, Gandy uses evident physical limitations of mechanisms to arrive at a general notion of machine computation; cf. section 4.

Gandy machines allow a direct and most straightforward formalization of paradigmatic parallel computations; it remains to be seen, whether they are also a convenient tool for the theoretical investigation of such computations. However, some results for Gandy's original formulation were obtained by Shepherdson and Dahlhaus & Makowski in 1988. In an independent second paper we will change the form of presentation from hereditarily finite sets to graphs, discuss connections to other models of parallel computation, and give some informative applications.

1. Structural classes & operations. Gandy machines consist of classes of "states" together with a "transition operation" that allows one to step from one state to the next. As mentioned above, we use Gandy's original set theoretic framework for the specification of such machines: states are represented by hereditarily finite sets, and transitions are given by restricted operations from states to states. The hereditarily finite sets are built up from a potentially infinite set U of atoms. These atoms can be thought of as the basic, unanalyzed components of machines. In this section and in section 2 we give

needed general set theoretic definitions; section 3 contains and motivates the formulation of two restrictive principles for mechanisms.

The hereditarily finite sets over U are obtained by the *finite powerset operation* Pp that does not admit the empty set as a subset:

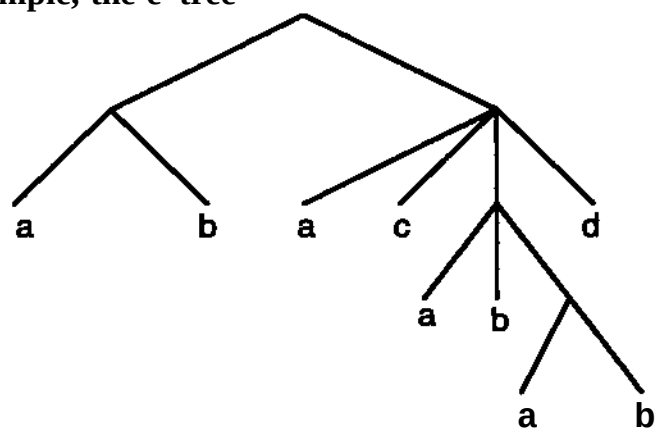
$$P_F(X)=\{Y \mid Y \subseteq X \ \&\ Y \neq 0 \ \&\ Y \text{ is finite}\};$$

the hierarchy of hereditarily finite sets is inductively defined by

$$HF_0 = 0$$

$$HF_{n+1} = P_F(HF_n \cup U)$$

The class HF of all hereditarily finite sets over U is the union of the HF_n's together with the empty set. The elements of HF are finite trees whose leaves are in U; for example, the e -tree



is a graphic representation of the set {{a, b}, {a, c, {a, b, {a, b}}, d}}. Here we are following a convention we will use throughout: we have a three-sorted language with names a, b, c, d,... for atoms from U and variables r, s, t, u,... ranging over U; the variables v, w, x, y, z,... range over HF and X, Y, Z,... over subsets of HF. In order to take into account finitely many different kinds of atomic components, one would have to increase the number of sorts; as this is without any theoretical consequence, we consider just the simplest case.

Two observations are immediate: (i) $0 \subseteq HF$, but for each $n \in N$, $0 \notin HF_n$; (ii) the hierarchy is cumulative. The e -relation on HF is well-founded; so we can use the *e* -recursion principle to define operations on HF. For example, the *transitive closure ofx,* Tc(x), is defined as $x \cup U\{Tc(y) \mid y \in x\}$; Tc(x) is the smallest transitive set y with $x \subseteq y$. The atoms contained in Tc(x) form the *support ofx,* Sup(x), i.e., Sup(x) = Tc(x)nU. Elements of HF, or *structures in HF,* are to reflect states of physical devices just as mathematical structures in physics are to represent states of nature. Thus, any isomorphic structure will

do as well; one should notice that this reflection is done somewhat indirectly, as only the e-relation is available.

Definition, x and y are e-*isomorphic* or simply isomorphic (x≡y) iff there is a bijection F: Tc(x)->Tc(y), such that for all zeTc(x): [for all w€Tc(z)(wez iff F(w)e F(z)) and for all re Sup(z): (re z iff F(r)e F(z))].

The class of structures isomorphic to a given structure x is called its *stereotype.* Notice that an e-isomorphism F between structures x and y is uniquely determined by a bijection *n* between Sup(x) and Sup(y) with F(r)=7c(r) for all re Sup(x). That comes from the fact that any permutation *n* on U can be extended to an e-isomorphism rc*:HF-»HF by e-recursion: rc*(x) = *{n(r)* I re x}u{rc*(y) I ye x}. We will write x^n for *n*(x).*

Definition. A subclass X of HF is called *structural* iff X is closed under e-isomorphisms; we use S as a variable over structural classes.

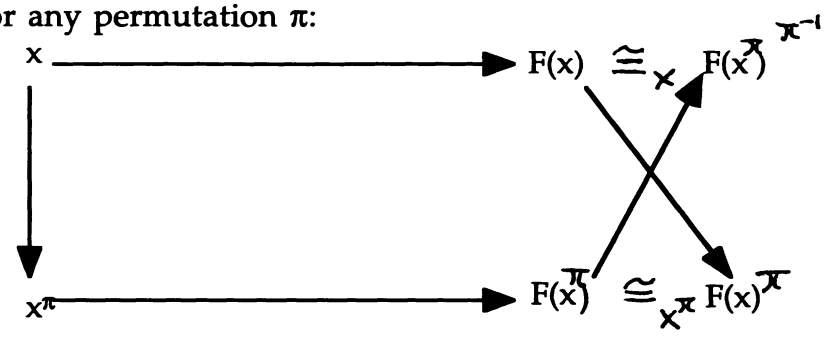
The set of states of a mechanical device is thus represented only by a class X of HF-structures that is structural. Gandy emphasized: "All the information about that state which is relevant to the operation of the machine must be encoded in any structure x which is used to describe it."

Transitions for a mechanical device must be operations F from a structural class S to S, where this class contains all the possible states of the device. What broad conditions should be satisfied by such an operation? — There is an obvious first requirement, namely, that isomorphic states are mapped to isomorphic states; but this is not enough, as the operation should depend only on the stereotype (i.e., "structure") of a state. Consider the operation F defined for the ordered pair <a, b> by F(<a, b>)={b} and for the ordered pair <c, d> by F(<c, d>)={c}; isomorphic states are indeed mapped to isomorphic states, but F operates not just on the structure of the states. To remedy this shortcoming by requiring full invariance, i.e., F(x*) = F(x)* for any permutation *n,* would be too restrictive: in stepping from one state to the next the device may be expanded, and additional atoms may be used for the new parts; there is no reason for restricting the choice of atoms (for building up new parts) so that identity results. To capture this idea we introduce a refined notion of e-isomorphism between structures in HF: x and y are called *e-*

isomorphic over A, $x \cong_A y$, where **A** is a set of atoms, iff there is a permutation π on **U** with $\pi(r)=r$, for all $r \in A$, and $x^\pi = y$. Using the abbreviation $x \cong_z y$ for $x \cong_{Sup(z)} y$, we can now formulate the informal requirement by $F(x^\pi) \cong_{x^\pi} F(x)^\pi$.

Definition. Let **S** be a structural class; an operation F: $S \to S$ is called *structural* iff for all permutations π and all $x \in S$: $F(x^\pi) \cong_{x^\pi} F(x)^\pi$.

More graphically, an operation is structural, if the following diagram commutes for any permutation π:

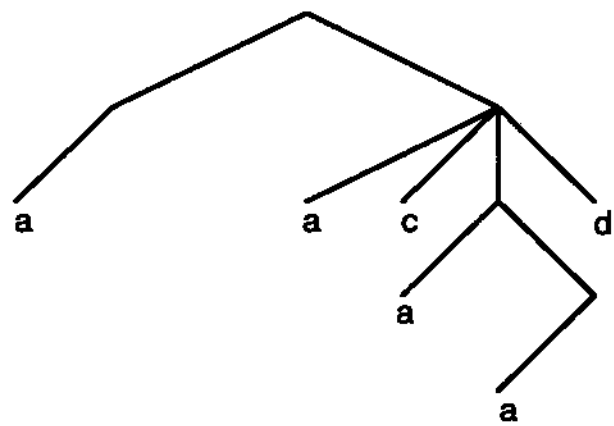


Now we have all the basic ingredients for the description of mechanical devices: a machine **M** is (described by) a pair <**S**, F>, where **S** is a structural class and F is a structural operation from **S** to **S**. If $x_0 \in S$ is an initial state of **M**, then $F(x_0)$, $F(F(x_0))$, ... are the subsequent states of **M**. Thus, **M** is a special case of a discrete dynamical system. Gandy pointed out that the class of machines given in this way is so comprehensive that any number theoretic predicate P can be decided by them. Let χ_P be the characteristic function of P and consider the following machine M_P with $S:=\{\{r\}^n \mid r \in U,\ n \geq 1\}$, where $\{r\}^1=\{r\}$ and $\{r\}^{n+1}=\{\{r\}^n\}$, and $F(\{r\}^n):= \{r\}^1$ if $\chi_P(n)=1$, else $\{r\}^2$; this machine does compute χ_P. Gandy used as one principle that the set theoretic rank of machine states is bounded; the "omniscient machine" is obviously not bounded in this sense. The principle is motivated by the fact that the hierarchical structure of actual machines, as reflected by the height of the epsilon-trees of their states, is not modified by their operation. The "principle of limitation of hierarchy" is then formulated as follows: if **M**= <**S**, F>, then the set theoretic rank of the states of **M** is bounded, i.e., there is a k in **N**, such that **S** is contained as a subset in HF_k. (Cf. end of section 2.) In the next two sections we will present physically motivated operating principles that will exclude such "omniscient" machines from being counted as Gandy machines; indeed, section 2 is preliminary, as it only formulates more precisely on what kind of structures operations are carried out.

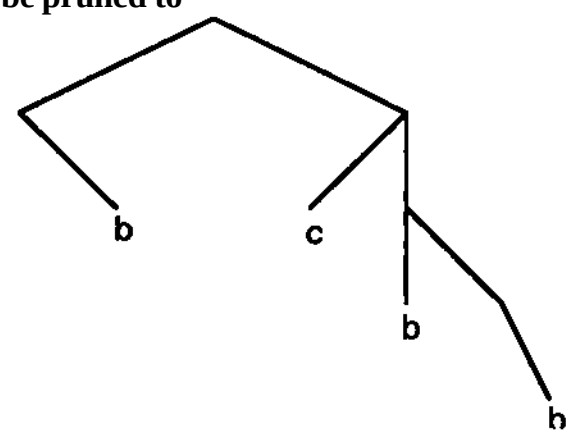
2. Substructures of states. Gandy's principle of unique assembly expresses, in his words, "that any device can be assembled from parts of bounded size, and that these parts can be so labelled that there is a unique way of putting them together. Model-construction kits aim, not always successfully, to satisfy this principle". We are not going to use that as a general restrictive principle for states of Gandy machines; however, we are using the notions involved in its formulation. So it is necessary to introduce some additional set theoretic preliminaries; in particular, we will need a "pruning operation" xtY for a variety of classes Y. As a first example we let Y be a subset of Sup(x) and define

$$xtY = (x \cap Y) \cup [\{y \in t(Y \cap Tc(y)) \mid y \in x\} \setminus \{0\}].$$

The set xtY is obtained from x by pruning all nodes with atoms that are *not* in Y (and resulting empty sets); i.e., we obtain the part of the e -tree that is built up solely from atoms in Y. We consider now the earlier G -tree; if Y = {a, c, d}, then xtY is



If Y = {b, c}, x will be pruned to



Three remarks are of immediate interest for our motivating considerations: (i) if Y=0, then =y coincides with =, (ii) if x=yyv then xtY=ytY, and (iii) if x=yy and Y is the support of x, then x=y.

As a second example for the pruning operation xtY we let Y be a subset of HF. Y may contain now elements y and z such that y is a subtree of z, i.e., ye Tc(z). To make the priming operation unique and prune with respect to the largest possible subtrees, we first define when Y is a *set of parts for* x, namely, just in case Pt(x,Y)=Y; the operation Pt(x,Y) is defined by *e* -recursion as follows:

$$Pt(x,Y)=(xnY) \cup \bigcup\{Pt(y,Y) \mid y \in x \ \&\ y \notin Y\}.$$

Note that Y is a set of parts for x iff (i) Y⊆Tc(x) and (ii) for all w, ze Y, if w*z, then weTc(z). The operation thus selects from Y those elements that are maximal in the € -tree for x. If every element of x is in Y, then Pt(x,Y)=x; in particular, Pt(x,x)=x. For a set Y of parts for x we define then as before

$$xtY = (xnY) \cup [\{yt(YnTc(y)) \mid y \in x\}\setminus\{0\}].$$

For sets of parts with Yi^Y2 we have xTYi=(xTY2)TYi. — One final auxiliary notion: Y *bars* x iff every *e* -chain r e xi *e* ... e x_n *e* x contains an element of Y. Now we can formulate the central notions for assembling machines.

Definition, (i) y is a *subassembly*[2] *for xfrom* Y, briefly y⊆*x, iff Y is a set of parts for x and y=xTY.

(ii) C is an *assembly for* x iff C is a set of subassemblies y for x from Y_y, such thatU{Y_y I yeCJbarsx.

An assembly for x is a covering for x in the literal sense of the word. Though elements of an assembly C for x are subassemblies for x, such a C can also be an assembly for y, y*x. To see that consider x = {{a, bi},..., {a, b_n}}, where a, *b*\, ..., b_n are distinct atoms; consider Y⊆Sup(x) with card(Y)≤n. If y = x u {{a}}, then yTY = xtY. Thus, x cannot be uniquely assembled from subassemblies of the form xtY with Y of properly smaller size, i.e., here YcSup(x). If C is an assembly for exactly one x, we say that x *can be uniquely assembled from* C or also that C *uniquely assembles to* w. If w=AX, we also say that C *uniquely assembles to* w *over* A.

[2] This is what Gandy calls a *located* subassembly. He suggested in his "Notes added in proof", p. 147, using these definitions, where an assembly is obtained from "located" subassemblies, instead of the definitions actually given in the body of the paper.

Gandy postulated a general boundedness condition for machine states: they can be uniquely assembled from subassemblies y of bounded size, where the *size ofy,* size(y), is understood as the cardinality of Sup(y): if M = <S, F>, then there is a qe N and for each xe S there is an assembly C for x, such that (1) x can be uniquely assembled from C, and (2) the size of every *ye* C is bounded by q. If M satisfies also the principle of limitation of hierarchy (cf. end of section 1), then the bound q is not just a bound for the size of each ye C, but it also determines a bound for the cardinality of each y: from the given less than q atoms only finitely many different sets can be built up at each level of HR_j, j≤k. I.e., card(y) <£, where the dots bounded by 2's indicate a stack of k two's. We are not using these restrictions; the relevant aspects follow from other considerations below, namely, the finiteness of structurally different parts that can be directly affected by a machine.

3. Locality of operations. Turing's requirement that a computation step should depend only on a bounded portion of the record was motivated by the evident limitations of the human sensory apparatus; as we mentioned already, memory limitations were viewed as the ultimate reason. This motivation is replaced here by an appeal to physical limitations: signals can only be conveyed with finite velocity; the possibility of instantaneous action at a distance is rejected by contemporary physics. This is the background for the *Principle of Local Causality,* formulated by Gandy in the following preliminary way:

The next state, Fx, of a machine can be reassembled from its restrictions to overlapping "regions" s and these regions are locally caused. That is, for each region s of Fx there is a "causal neighborhood" f2Tc(x) of bounded size such that Fxts depends only on xtt.

We are going to distinguish between the "local determination of regions" and "assembling the next state" (unique up to isomorphism) from these regions. That regions are locally determined means that there is some structural function G acting on suitable substructures of x, but also that G's domain is finite up to isomorphism. The physical motivation for the domain's finiteness is straightforward[3]: within the bounds set by the propagation of signals with finite velocity there can be only finitely many atoms, as it is assumed that their size has a lower bound. (Cf. section 3, Physical Limits, of [Mundici & Sieg].)

[3] Gandy derives this finiteness by the considerations mentioned at the end of section 2. Our direct appeal to physical constraints at this point avoids the rather forced detour through the particular set theoretic frame.

Now let T be the finite set of stereotypes whose elements make up the domain of G. The causal neighborhoods of a state x are structurally determined by T. y is called a *T-maximal subassembly ofx* just in case ye UT, y is a subassemby of x, and there is no subassembly y* of x in UT, such that y is e-embeddable in y*.[4]

Definition. Cn(x) = {y I y is a *T-maximal subassembly ofx};* Cn(x)is called the *set of causal neighborhoods ofx* determined by the set T of stereotypes.

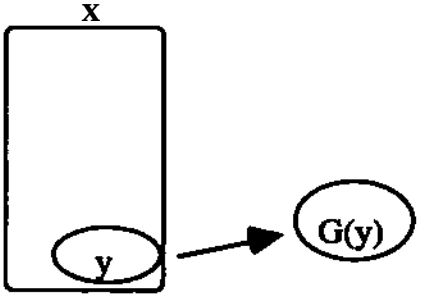
The underlying informal idea is that the causal neighborhoods y of x determine unique regions G(y) via the structural operation G; these regions are isomorphic over y to subassemblies v of F(x), i.e., $v=_y G(y)$. G may introduce additional atoms; we require that these atoms be new for x, i.e., that G satisfy Sup(G(y))nSup(x)£Sup(y). For such G the isomorphism between v and G(y) can be strengthened to $v=_x G(y)$. (This is going to be used after the next definition to specify in what sense the "effects of causes" are unique.)

Definition. Dr(x)={v I (3y)(yeCn(x) & v=G(y))}; Dr(x) is called the *set of determined regions of x* — the regions are determined by G from elements in Cn(x).

Viewing the causal neighborhoods of x as causes and the determined regions as effects, we can formulate that "every cause has an effect":

(1) (VyeCn(x)) (3veDr(x)) v=G(y),

Diagrammatically, the relations are as follows:



[4] I.e., y is e -isomorphic to a proper subtree of y^s e -tree.

If determined regions are isomorphic over $\mathbf{x}$, we actually require -- to prevent the construction of omniscient machines (cf. Gandy's discussion on pp. 139-40) -- that they are identical:

$$(2)\ (\forall \mathbf{v}, \mathbf{w} \in Dr(\mathbf{x}))\ (\mathbf{w} \cong_{\mathbf{x}} \mathbf{v} \rightarrow \mathbf{w}=\mathbf{v}).$$

Now we have to take the second step: having determined the regions locally, we must assemble (an isomorphic copy of) the next state from them. To see the issues clearly, consider a simple example: if $\mathbf{x} = \{\langle \mathbf{a}, \mathbf{b}\rangle, \langle \mathbf{c}, \mathbf{d}\rangle\}$, then we might have chosen $G(\langle \mathbf{a}, \mathbf{b}\rangle) = \{\mathbf{b}, \mathbf{f}\}$ and $G(\langle \mathbf{c}, \mathbf{d}\rangle) = \{\mathbf{d}, \mathbf{g}\}$. In this case $F(\mathbf{x}) = \{\mathbf{b}, \mathbf{f}, \mathbf{d}, \mathbf{g}\}$; alternatively, we might have chosen $G(\langle \mathbf{c}, \mathbf{d}\rangle) = \{\mathbf{d}, \mathbf{f}\}$, in which case $F(\mathbf{x}) = \{\mathbf{b}, \mathbf{f}, \mathbf{d}\}$. F is a *functional* operation, as the computations are to be deterministic; thus, such ambiguity must not exist. Supposing that overlaps do not occur would be too restrictive. Indeed, multiple causal neighborhoods do sometimes introduce the same new atoms: consider, for example, the outward growth of the underlying grid of a cellular automaton. The agreement or disagreement of the newly introduced atoms must be predetermined; i.e., the isomorphism type (over $\mathbf{x}$) of any class of determined regions with common new elements must also be given by a function on appropriate substructures of $\mathbf{x}$. Calling such a function G_2, we require that it satisfy the same restrictions as G, except that regions determined by it need only be unique up to isomorphism over $\mathbf{x}$ (as we have introduced G_2 explicitly for determining only isomorphism types over $\mathbf{x}$, whereas G is required to produce a particular assembly). We use G_1 now for G; if i=1 or 2 and T_i is the set of stereotypes of states in the domain of G_i, then Cn_i and Dr_i are the causal neighborhoods and regions determined by G_i and T_i. (The principles are formulated for fixed $\mathbf{M} = \langle \mathbf{S}, \mathbf{F}\rangle$, G_i and T_i.)

Principle I (Local Causation). For every $\mathbf{x} \in \mathbf{S}$:

$$(1.1)\ (\forall \mathbf{y} \in Cn_1(\mathbf{x}))\ (\exists \mathbf{v} \in Dr_1(\mathbf{x}))\ \mathbf{v}=G_1(\mathbf{y}),$$

$$(1.2)\ (\forall \mathbf{y} \in Cn_2(\mathbf{x}))\ (\exists \mathbf{v} \in Dr_2(\mathbf{x}))\ \mathbf{v}=G_2(\mathbf{y}),$$

and

$$(2)\ (\forall \mathbf{v}, \mathbf{w} \in Dr_1(\mathbf{x}))\ (\mathbf{w} \cong_{\mathbf{x}} \mathbf{v} \rightarrow \mathbf{w}=\mathbf{v}).$$

An arbitrarily large number of regions determined by G_1 might introduce some common atom, but the size of the regions of G_2 are bounded. It is a combinatorial fact established by Gandy (on pp. 143-5 of his [1980]), that $F(\mathbf{x})$ is determined uniquely up to isomorphism as long as the regions of G_2

determine the overlap on up to r_{Gl}-many regions obtained via G! at a time, where

$$r_{Gl} = \max\text{Icardf}\text{Supf}G\hat{}y)) \setminus \text{Sup}(y)) \ I \ y \ e \ T\hat{}+l.$$

The G_1-determined regions of x and the G2-structurally determined overlaps should allow us to put together F(x) - up to isomorphism over x.

Principle II (Unique Assembly). For every xe S (writing r for r_{Gl}):

1. (VCQ^x)) [card(C)≤r & Pi{Sup(v) I veC} is not a subset of Sup(x)
-> (3weDr$_2$(x))(VveC)v$\underline{c}$*w]
2. Dri(x) uniquely assembles to F(x) over x.

We are finally where we wanted to be: a machine M = <S, F> is called a *Gandy Machine* if and only if there are finite sets Ti and T2 of stereotypes and structural functions Gi and G2 so that Principles I and II are satisfied.

To make this rather abstract discussion a bit more textured, we represent a Turing machine M as a Gandy machine; we think of a Turing machine here as a Post production system as in [Davis]. Suppose the set of symbols of M to be {so,..., Sk} and that of internal states to be {qo,..., q_r}- We can represent so,../Sk/<lO/—Air by $\{r\}^1$,...,$\{r\}^{r+k+2}$ respectively, for every atom r. A cell a of the tape which contains the symbol s and has cell b to its immediate right is indicated by the ordered pairs <a, b> and <s, a>. (We will ignore the problem of keeping track of the leftmost and rightmost cell; we can assume that extra symbols are introduced, as was done by Post.) The machine's head position and its internal state qi are given by <qi, a>, where a is the cell currently being read. A state of a Turing machine is its instantaneous description which can be represented by the set of all the ordered pairs for cells together with the pair indicating the head position and internal state; let S be the class of all such states.

We can define Gi as follows. For the command "if in state qi and scanning symbol Sm, move to the right and enter state qj", we have

$$G_1(\{\langle q_i, a\rangle, \langle s_m, a\rangle, \langle a, b\rangle\}) = \{\langle q_j, b\rangle, \langle s_m, a\rangle, \langle a, b\rangle\}$$

and similarly for the other commands. In addition we have for each symbol s in the language, each internal state q, and each c≠a

$$G_1(\{\langle q, c\rangle, \langle s, a\rangle, \langle a, b\rangle\}) = \{\langle s, a\rangle, \langle a, b\rangle\}$$

Again, these are made for all atoms a and b (and all atoms r underlying the representations of s and q), i.e., we take the "structural closure" of the

function minimally indicated above. Since at most one neighborhood introduces a new cell, we can choose G2 = Gi. These define a unique structural function F on S in accordance with the principles above, and F clearly carries out the same transfomations on instantaneous descriptions as the given Turing machine M.

4. Gandy's Thesis. Gandy analyzed the notion of machine computation or, making the first analytic step, what can be done by a "discrete deterministic mechanical device". In this Gandy followed Turing's methodological ways and sharpened the informal notion at hand. First of all, he considered, as Turing did, calculations that are deterministic and whose progress can be "described in discrete terms". Secondly, he made explicit broad physical considerations that constrain a mechanism in carrying out such calculations, namely, (i) there is a lower bound on the size of the machine's atomic parts, and (ii) signals can be transmitted only with bounded velocity. He considered the possibility of carrying out steps in parallel as distinctive for such devices and appealed to (i) and (ii) to motivate restrictive principles. Gandy did not present yet another model for parallel computations, but rather attempted to isolate principles *any* such model must obey.[5]

The fact that M = <S, F> is a "discrete dynamical system" reflects simply the discrete and deterministic character of the devices that are being modeled: discrete states can be adequately represented by hereditarily finite sets[6]; transitions must be given by an operation, as the calculations are to be deterministic. (Since the atoms are indistinguishable, operations must depend only on the structure of states; thus the insistence that S be a structural class and F a structural operation.) It is only now that the specifically physical considerations come into play. By (i) and (ii) the devices can directly affect just finitely many structurally different parts of a state; however, as they can operate in parallel, the total transition must be uniquely determined by local actions on such necessarily bounded parts.

In sum then, *Gandy's Thesis* is the claim that any discrete deterministic mechanical device (can be represented as a discrete dynamical system and its

[5] To see very dearly, how directly Gandy's analysis "follows" Turing's, the reader should look at our earlier work on K-graph machines that generalized, in a suitable sense, Turing's; cf. [Sieg *6c* Byrnes 1996 and 1998] and references mentioned there, in particular, [Turing 1953] and [Kolmogorov & Uspenski].

[6] Graphs would work as well, if not better, as structural features of the devices can be more directly reflected; indeed, we have developed a simple graph-theoretic presentation; cf. [Byrnes & Sieg].

operation) must satisfy the restrictive principles of local causation and unique assembly. This is completely parallel to Turing's Thesis expressing the claim that any human computor (can be represented by a discrete dynamical system and his operation) must obey appropriate locality conditions. The crucial difference between these two abstract models of computation lies in the fact that Turing machines modify only one bounded part of a state, whereas Gandy machines operate on arbitrarily many bounded parts.

References

Byrnes, John
1993 Notions of computation; M.S. Thesis, Department of Philosophy, Carnegie Mellon University, Pittsburgh.

Byrnes, John and Wilfried Sieg
1996 A graphical presentation of Gandy's parallel machines (Abstract), Bulletin of Symbolic Logic, 452-3.

Dahlhaus, Elias and Johann Makowski
1988 Gandy's principles for mechanisms as a model of parallel computation; in: *The Universal Turing Machine* - A Half-Century Survey (R. Herken, ed.), Oxford University, 309-314.

Davis, Martin
1958 *Computability and Unsolvability;* McGraw-Hill, New York.

Gandy, Robin
1980 Church's Thesis and principles for mechanisms; in: *The Kleene Symposium* (J. Barwise, H.J. Keisler, K. Kunen, eds.), North-Holland Publishing Company, 123-148.

Lamport, Leslie and Nancy Lynch
1990 Distributed Computing: Models and Methods; in: *Handbook of Theoretical Computer Science Q.* van Leeuwen, ed.), Elsevier Science Publisher B.V.

Mundici, Daniele and Wilfried Sieg
1995 Paper Machines; Philosophia Mathematica 3 (1), 5-30.

Shepherdson, John
1988 Mechanisms for computing over arbitrary structures; in: *The Universal Turing Machine* - A Half-Century Survey (R. Herken, ed.), Oxford University, 581-601.

Sieg, Wilfried
1990 Gandy Machines; Manuscript
1994 Mechanical procedures and mathematical experience; in: *Mathematics and Mind,* A. George (ed.), Oxford University Press, 71-117.

Sieg, Wilfried and John Byrnes

19% K-graph machines: generalizing Turing's machines and arguments; in: *Gödel '96* (P. Hájek, ed.), Lecture Notes in Logic 6, Springer Verlag, 98-119.

1998 Gödel, Turing and K-graph machines; to appear in: *Logic in Florence* (A. Cantini, E. Casari, P. Minari, eds.), Kluwer Academic Publishers.

Turing, Alan

1936 On computable numbers, with an application to the *Entscheidungsproblem;* Proc. London Math. Soc. Ser. 2, 42, 239-65.