ACTAS DEL VI CONGRESO
DR. ANTONIO A. R. MONTEIRO
(2001), Páginas 129–136

Interaction machines and System Theory

Manuel Fidel
Department of Computer Science, Universidad Nacional del Sur
Av. Alem 1253, (8000) Bahía Blanca, Argentina
ccfidel@criba.edu.ar
Martín Figallo
Departamento de Matemática, Universidad Nacional del Sur
Av. Alem 1253, (8000) Bahía Blanca, Argentina

Abstract

In 1956, R. Ashby considered as models of dynamic systems certain functions we shall call Ashby transformations. In this paper we prove that the interaction machines defined by P. Wegner in 1996 are equivalent in their expresive potential to particular Ashby transformations, thus showing that there exists a relationship between two disciplines which, up to now, have seemed to ignore each other. They are the theory of computer science and system theory, the latter being capable of becoming a basic part of the former.

1 Introduction

Computational technology has developed significantly in its theoretical fundamentals since its birth up to the present.

The notion of algorithm arose in response to foundational questions in mathematics. Computer science, which arose with de advent of computers, adopted this notion as fundamental paradigm.

Electronic digital computers appeared in 1945. They computed one step at a time and they were limited to a finite number of steps and to a finite number of instructions. These computers are the embodiment of the Turing machine. But, at the present, computer science deal whith important process expresions which do not qualify as algorithms [5].

An operating system, similar to any nonterminating reactive processes cannot be modelled by algorithms. The same happens with concurrent processes.

The development of a conceptual framework and formal theoretical foundation for object-oriented programming has proved to be a hard task owing to the fact that object observable behaviour cannot be expressed by means of an algorithm.

The algorithmic specifications of certain problems, in which it is necessary to handle interactively variable factors, are many and quite complex, supposing that such specifications exist. Therefore, algorithmic computation is insufficient to model interactive behaviour.

In the search of a more powerful computing mechanism, capable of modeling present-day computers, P. Wegner introduced interaction machines [9]. They are simple extensions of Turing machines, but they have both observable behaviour and formal properties which are completely different from those of Turing machine.

Wegner proved that the observable behaviour of interaction machines, unlike that of Turing machines cannot be specified by mathematical models.

Interaction machines can model the behaviour of objects in object-oriented programs, since interaction is the key notion which makes object behaviour more powerful than that of procedures.

Turing machines have a limited power because they compute an output from an initial input on the tape and cannot interact with an external environment during a computation.

This characteristic of Turing machines is the same that von Bertalanffy, founder of general system theory, observed in the systems studied by physicist. von Bertalanffy noticed that these systems were closed: they do not interact with the outside world, that is to say with the context. When physicist makes a model (for example, a model of an atom), he assumes that all the mass, particles and forces that act or affect the system are included in the model. This makes it possible to calculate future states with perfect precision since all the necessary information is known. However, as a biologist, von Bertalanffy knew that supposing such a thing was imposible for the majority of practical phenomena.

He noticed that real systems are open, in the sense that they interact with their environments and can acquire new properties, thus resulting in a continuous evolution. This characteristic of real systems is present in computational systems too.

2 Preliminaries

In this section we shall review some basic notions and some result on formal languages, phrase structure grammars (or type 0 grammars), Turing machines and Ashby transformations. For further details, see [2], [3], [4], [6], [7], y [8].

In the remaining part of the work we shall denote by $\mathbb{Z}$ y $\mathbb{N}$ the sets of integers and positive integers respectively.

Let us recall that a non-empty set Σ is an alphabet if for every $x \in \Sigma$, x is an indivisible symbol, i.e. if x is not made up of subsymbols such that they are in turn elements of Σ.

A word defined on the alphabet Σ is any finite sequence of symbols of Σ. Besides, if

w is a word defined on Σ, the length of w is the number $long(w)$ of ocurrences of symbols of Σ in w.

We shall denote by Σ^* the set of all words of finite length defined on Σ.

A language defined on Σ is any subset L of Σ^*.

A phrase structure grammar G is a quadruple $G = (V_n, V_t, S, P)$ where:

- V_t is a non-empty set of terminal symbols (an alphabet),
- V_n is a non-empty set disjointed with V_t of non-terminals symbols,
- $S \in Vn$ is the initial symbol,
- P is a finite set of rules of production of the form $\alpha \rightarrow \beta$ where $\alpha \in (V_n \cup V_t)^* \cdot V_n \cdot (V_n \cup V_t)^*$ and $\beta \in (V_n \cup V_t)^*$.

In addition:

- $\varphi\alpha\mu$ derives directly into $\varphi\beta\mu$ according to G and we shall denote it by $\varphi\alpha\mu \overset{G}{\rightarrow} \varphi\beta\mu$, if there exists a rule of production $\alpha \rightarrow \beta \in P$,
- α derives into β and we shall denote it $\alpha \overset{G+}{\rightarrow} \beta$, if there exists $\gamma_1, \gamma_2, \dots, \gamma_n \in (V_n \cup V_t)^*$ such that $\alpha \overset{G}{\rightarrow} \gamma_1$, $\gamma_1 \overset{G}{\rightarrow} \gamma_2, \dots, \gamma_n \overset{G}{\rightarrow} \beta$.

Let $G = (V_n, V_t, S, P)$ be a phrase structure grammar, then:

- The language generated by G is the set $L(G) = \{w \in V_t^* : S \overset{G+}{\rightarrow} w\}$.
- If $\alpha \in (V_n \cup V_t)^*$, the set of all the derivations of α according to G is the set $D_G(\alpha) = \{w \in (V_n \cup V_t)^* : \alpha \overset{G+}{\rightarrow} w\}$.

A Turing machine is a tuple $T = (S, \Sigma, \delta, s_0, F)$, where:

- S is a set of states, $S \neq \emptyset$;
- Σ is an alphabet;
- δ is a partial function with domain in $S \times \Sigma$ and image in $S \times \Sigma \times \{I, D, N\}$;
- s_0 is the initial state, $s_0 \in S$;
- F is the set of final states, $F \subseteq S$.

A configuration of a Turing machine $T = (S, \Sigma, \delta, s_0, F)$ is a triple $(s, \alpha, i) \in S \times \Sigma^* \times I\!N$.

We shall call transition of T any pair of the binary relation $\vdash$, between configurations, defined : $(s, \alpha a\beta, i) \vdash (s', \alpha b\beta, i')$ if, and only if $\delta(s, a) = (s', b, M)$ where $s, s' \in S$; $\alpha, \beta \in \Sigma^*$; $M \in L, N, R$; $a \in \Sigma$ is the i–nth symbol of the word $\alpha a\beta$ and

$$i' = \begin{cases} i-1 & si \quad M = L \\ i & si \quad M = N \\ i+1 & si \quad M = R \end{cases}$$

A language accepted by a Turing machine $T = (S, \Sigma, \delta, s_0, F)$ is a language $L(T) = \{w \in \Sigma^* : (s_0, w, 1) \vdash^* (s_f, \alpha, i)$ for some $s_f \in F, \alpha \in \Sigma^*, i \in \mathbb{Z}\}$.

The well-known Theorem 2.1 shows the equivalence between phrase structure grammars and Turing machines as far as expressive power is concerned.

Theorem 2.1 *Let L be a language defined on an alphabet Σ. Then, the following conditions are equivalent:*

(i) *L is generated by a phrase structure grammar G,*

(ii) *there exists a Turing machine T such that L is accepted by T.*

Interaction machines were introduced by P. Wegner as a simple extension of Turing machines. An interaction machine is a Turing machine with input actions that allow the input of external data during the computational process.

In 1956, R. Ashby considered, in [1], certain notions which can be presented as follows.

Let S be a non-empty set whose elements we shall call states, then:

(a) a transition is a pair $(\alpha, \beta) \in S \times S$.

(b) a transformation T is a binary relation defined over S and an initial state of T is a fixed element of S.

(c) a transformation T is closed if the domain of T is S and is one-valued if T is a function of S in S.

3 Interaction machines and Asbhy's transformatons

First we shall prove that the expressive power of Ashby's transformations is at least equal to the expressive power of phrase structure grammars.

Clearly, the observable behaviour of a phrase structure grammar is given by $L(G)$, the language generated by G.

In order to prove that an Ashby transformation has at least as much expressivness as a phrase structure grammar, we must establish what we understand by expressive power, or observable behaviour, of an Ashby transformation.

In [1] Ross Ashby says : " The series of positions taken by the system in time clearly corresponds to the series of elements generated by the succesive powers of the transformation. Such a sequence of states defines a trajectory or line of behaviour."

For this reason, given an Ashby transformation T, its observable behaviour will be determined by the sequence $T(s), T^1(s), \ldots T^n(s), \ldots$, s being the initial state of T.

Definition 3.1 *Let $G = (V_n, V_t, S, P)$ be a phrase structure grammar. We shall call Ashby transformation associated with G the function defined on the family of subsets of $(V_n \cup V_t)^*$, by means of the following prescriptions. We shall represent it with T_g.*

(i) $T_g(\emptyset) = \emptyset$,

(ii) $T_g(\{\alpha\}) = \{\alpha\}$ *if* $\alpha \in V_t^*$,

(iii) $T_g(\{\alpha\}) = D_G(\alpha)$ *if* $\alpha \notin V_t^*$,

(iv) $T_g(\{\alpha_1, \alpha_2, \ldots, \alpha_k\}) = D_G(\alpha_1) \cup D_G(\alpha_2) \cup \ldots \cup D_G(\alpha_k)$, *where* $\alpha_1, \alpha_2, \ldots, \alpha_k \in (V_n \cup V_t)^*$

The lemma that follows will be used later.

Lemma 3.1 *Let* $\beta \in (V_n \cup V_t)^*$ *and* $k \in I\!N$. *Then these conditions are equivalent.*

(i) $\beta \in T_g^{k+1}(\{S\})$,

(ii) *there exists* $\alpha \in T_g^k(\{S\})$ *such that* $\alpha \xrightarrow{G} \beta$.

Proof
(i) $\Longrightarrow$ (ii)

(1) $\alpha \xrightarrow{G} \beta$,

(2) $\alpha \in T_g^k(\{S\})$,

(3) $D_G(\alpha) \subseteq T_g(T_g^k(\{S\})) = T_g^{k+1}(\{S\})$ [def. 3.1, (2)]

(4) $\beta \in D_G(\alpha)$, [(1)]

(5) $\beta \in T_g^{k+1}(\{S\})$. [(3), (4)]

(ii) $\Longrightarrow$ (i)

(1) $\beta \in T_g^{k+1}(\{S\})$,

(2) $\beta \in T_g(T_g^k(\{S\}))$, [(1)]

(3) there exists α such that $\alpha \in T_g^k(\{S\})$ and $\beta \in D_G(\alpha)$, [(2) and def. 3.1]

(4) $\alpha \xrightarrow{G} \beta$ [3] ∎

Now we shall prove the most important result in this paper.

Theorem 3.1 *Let L be the language generated by a phrase structure grammar; then there exists an Ashby transformation such that the language generated by G and the observable behaviour of T coincide.*

Proof

Let $G = (V_n, V_t, S, P)$ be a phrase structure grammar and T_g the Ashby transformation associated to G.

Let us see that G and T have the same observable behaviour. In other words, we shall prove that for every $w \in V_t^*$ these conditions are equivalent:

(i) $w \in L(G)$

(ii) there exists $k \in \mathbb{N}$ such that $w \in T^k(\{S\})$

(i) $\Longrightarrow$ (ii)

(1) $w \in L(G)$

(2) $S \overset{G+}{\rightarrow} w$, $w \in V_t^*$ [(1)]

(3) there exists $\gamma_1, \gamma_2, ..., \gamma_n \in (V_n \cup V_t)^*$ such that $\alpha \overset{G}{\rightarrow} \gamma_1$, $\gamma_1 \overset{G}{\rightarrow} \gamma_2$, ..., $\gamma_n \overset{G}{\rightarrow} w$ [(2)]

(4) $\gamma_1 \in D_G(S) \subseteq T(\{S\})$ [(3)]

(5) $\gamma_2 \in T^2(\{S\})$ [(3), (4) and lemma 3.1]

Going on with the procedure we prove that :

(6) $\gamma_n \in T^n(\{S\})$

(7) $w \in T^{n+1}(\{S\})$ [(3), (5) and lemma 3.1]

(ii) $\Longrightarrow$ (i)

(1) there exists $k \in \mathbb{N}$ such that $w \in T^k(\{S\})$

(2) there exists $\gamma_1 \in T^{k-1}(\{S\})$ such that $\gamma_1 \overset{G}{\rightarrow} w$ [(1) and lemma 3.1]

(3) there exists $\gamma_2 \in T^{k-2}(\{S\})$ such that $\gamma_2 \overset{G}{\rightarrow} \gamma_1$ [(2) and lemma 3.1]

going on with the procedure we prove that

(4) there exists $\gamma_{k-1} \in T(\{S\})$ such that $\gamma_{k-1} \overset{G}{\rightarrow} \gamma_{k-2}$

(5) $S \overset{G}{\rightarrow} \gamma_{k-1}$ [(4)]

(6) $S \overset{G+}{\rightarrow} w$ [(2), (3), (4), (5)]

(7) $w \in L(G)$ [(6)] ■

The following corollary is deduced from Theorem 3.1 and the equivalence that exists between the Turing machines and phrase structure grammars with respect to their expressive power.

Corolary 3.1 *Let L be a language accepted by a Turing machine M, then there exists an Ashby transformation such that the language accepted by M and the observable behaviour coincide.*

After introducing transformations as representations of any mechanism, R. Ashby noticed the fact that a machine (or a system) may act under different conditions and that therefore it can change the way it behaves.

With the purpose of finding an appropriate representation for a machine that changes its behaviour, this depending on external factors, the notion of parameter was introduced.

If we have a machine that behaves differntly according to external factors, then different transformations will be used to represent the different ways of behaving. In addition to them, a parameter will be used, the value of which will indicate the transformation to be applied.

A real machine whose behaviour can be represented by such a set of transformations is called by Ashby a machine with input. The set of transformations is its canonical representation and the parameter, as something that can vary, is its input.

This parameter is what allows Ashby transformations not to be closed to the outside world; on the contrary, they interact with it.

We consider that now the relationship between Wegner's interaction machine and Ashby transformations with parameter is clear. By Corolary 3.1 we know that any Turing machine is equivalent to some Ashby transformation as far as observable behaviour is concerned. The input operations are to interaction machines as parameters are to Ashby transformations. They are what, in both cases, allow them to interact with external environment.

These arguments justify the theorem that follows:

Theorem 3.2 *Let M be an interaction machine, then there exists an Ashby transformation with parameter T such that the observable behaviour of M and T coincide.*

4 Conclusions

The theoretical fundamentals of computer science have evolved from models equivalent to the Turing machine to P. Wegner's interaction models, which are particular cases of the dynamic systems of the general system theory.

References

[1] R. Ashby, *An Introduction to Cybernetics*, Chapman & Hall Ltd and Univ. Paperbacks, 1971.

[2] Y. Bar-Hillel, M. Perles and E. Shamir, On formal properties of simple phrase structure grammars, *Languages and Information*, pages 116-150, 1964.

[3] N. Chomsky, Three models for the description of language, *IEEE- Transactions on Information Theory*, 3(2):113-124, 1956.

[4] N. Chomsky, On certain formal properties of grammars, *Information and Control*, 2(2):137-167, 1959.

[5] K. Fant, " A critical review of the notion of the algorithm in computer science ", 1997.

[6] Hopcroft and Ullman, *Formal languages and their relations to autómata*, 1969.

[7] Hopcroft and Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison Wesley Publishing company, Massachussets, California, 1979.

[8] Minsky, *Computation: Finite and infinite machines*, Prentice Hall, Ney Jersey, 1967.

[9] P. Wegner, "The Paradigm Shift from Algorithms to Interaction", Brown University, Final Draft, October 14 1996.