On Representations of Problems of Reasoning about Actions

Saul Amarel
RCA Laboratories
Princeton, N.J.

1. INTRODUCTION

The purpose of this paper is to clarify some basic issues of choice of representation for problems of reasoning about actions. The general problem of representation is concerned with the relationship between different ways of formulating a problem to a problem solving system and the efficiency with which the system can be expected to find a solution to the problem. An understanding of the relationship between problem formulation and problem solving efficiency is a prerequisite for the design of procedures that can automatically choose the most 'appropriate' representation of a problem (they can find a 'point of view' of the problem that maximally simplifies the process of finding a solution).

Many problems of practical importance are problems of reasoning about actions. In these problems, a course of action has to be found that satisfies a number of specified conditions. A formal definition of this class of problems is given in the next section, in the context of a general conceptual framework for formulating these problems for computers. Everyday examples of reasoning about actions include planning an airplane trip, organizing a dinner party, etc. There are many examples of industrial and military problems in this category, such as scheduling assembly and transportation processes, designing a program for a computer, planning a military operation, etc.

The research presented in this paper was sponsored in part by the Air Force Office of Scientific Research, under Contract Number AF49(638)-1184. Part of this work was done while the author was on a visiting appointment at the Computer Science Department of the Carnegie Institute of Technology, Pittsburgh, Pa. At Carnegie Tech. this research was sponsored by the Advanced Research Projects Agency of the Office of the Secretary of Defense under Contract Number SD-146.

We shall analyze in detail a specific problem of transportation scheduling – the 'missionaries and cannibals' problem (which is stated in section 3) – in order to evaluate the effects of alternative formulations of this problem on the expected efficiency of mechanical procedures for solving it, and also in order to examine the processes that come into play when a transition takes place from a given problem formulation into a better one. After the initial verbal formulation of the missionaries and cannibals problem in section 3, the problem undergoes five changes in formulation, each of which increases the ease with which it can be solved. These reformulations are discussed in sections 4 to 11. A summary of the main ideas in the evolution of formulations, and comments on the possibility of mechanizing the transitions between formulations are given in section 12.

2. PROBLEMS OF REASONING ABOUT ACTIONS

A problem of reasoning about actions (Simon, 1966) is given in terms of an initial situation, a terminal situation, a set of feasible actions, and a set of constraints that restrict the applicability of actions; the task of the problem solver is to find the 'best' sequence of permissible actions that can transform the initial situation into the terminal situation. In this section, we shall specify a *system of productions*, P, where problems of reasoning about actions can be naturally formulated and solved.

In the system P, a basic description of a situation at one point in time is a listing of the basic features of the situation. The basic features are required for making decisions about actions that can be taken from the situation. We call a situation a *state of nature* (an N-state). The language in which N-states are described is called an *N-state language*. Such a language is defined by specifying the following:

(i) a non-empty set U_0 called the *basic universe*; this set contains the basic elements of interest in situations (the individuals, the objects, the places);

(ii) a set of basic predicates defined for elements of U_0 (properties of elements and relations between elements);

(iii) a set of rules of formation for expressions in the language.

The rules of formation determine whether an N-state language is a linear language, a two-dimensional (graphic) language, or it has some other form. Regardless of the form taken by an expression in an N-state language, such an expression is meant to assert that a given element in U_0 has a certain property or that a given subset of elements in U_0 are related in a specified manner. Thus, an expression in an N-state language has the logical interpretation of a true proposition about a basic feature of the situation. A finite set (possibly empty) of expressions in an N-state language is called a *configuration*. The empty configuration will be written Λ. In the logic interpretation, a (non-empty) configuration is a conjunction of the true assertions made by its component expressions. The set union of two configurations is itself a configuration. If α and β are configurations, then their union will be written α, β. A *basic description*, s, of an N-state is a configuration from which all true statements about the N-state (that can be expressed in the terms of the N-state language) can be directly obtained or derived. Thus a basic description completely characterizes an N-state. Henceforth we shall refer to an N-state by its basic description.

A derived description of an N-state at one point in time is a listing of compound features of the N-state. Compound features are defined in terms of the basic features, and they are intended to characterize situations in the light of the problem constraints, so that decisions about the legality of proposed actions can be made. We denote by $d(s)$ a derived description that is associated with an N-state s. The language in which derived descriptions are formulated is an extension of the N-state language, and it is called the *extended description language*. Such a language is defined by the following:

(i) a set U_1 called the *extended universe*, where $U_0 \subset U_1$ (this is not necessarily a proper inclusion); the extension of U_0 contains compound elements of interest (definable in terms of the basic elements in U_0), and possibly new elements (not obtainable from U_0) that are used for building high level descriptions;

(ii) a set of new predicates defined for elements of U_1 (properties and relations that are required for expressing the constraining conditions of the problem);

(iii) a set of rules of formation for expressions in the language.

The rules of formation in this language are identical with those of the N-state language. Each expression in the extended description language has the logical interpretation of a proposition about a compound feature in a situation. A derived description $d(s)$ is a set of expressions in the extended description language (it is a configuration in the language). In the logical interpretation, $d(s)$ is a conjunction of the propositions that are specified by its constituent expressions.

The *rules of action* in the system P specify a possible next situation (next in time with respect to a given time scale) as a function of certain features in previous situations. The complexity of a problem about actions is determined by the nature of this dependence. There is a sequential and a local component in such a dependence. The sequential part is concerned with dependencies of the next situation on features of sequences of past situations. We will not be concerned with such dependencies in this paper. The local part is concerned with the amount of local context that is needed to determine a change of a basic feature from one situation to the next.

In the specification of a rule of action, an N-state is given in terms of a *mixed description* s', which is written as follows:

$$s' = s;\ d(s), \tag{2.1}$$

where s is the basic description of the N-state, and $d(s)$ is its associated derived description. Let A be a feasible action and let (A) denote the rule of action that refers to A. A rule of action is given as a transition schema between mixed descriptions of N-states, and it has the following form:

$$(A):\ s_a;\ d(s_a) \rightarrow s_b;\ d(s_b) \tag{2.2}$$

The feasible action A is defined as a transformation from the N-state s_a to the N-state s_b. If A is applied at s_a, then the next N-state will be s_b. The rule (A) specifies the condition under which the application of A at s_a is permissible. This is to be interpreted as follows: 'If $d(s_a)$ and $d(s_b)$ are both satisfied, then the application of A at s_a is permissible.' A derived description $d(s)$ is satisfied if it is true under the logical interpretation. The rule (A) imposes a restriction on the mapping $A: s_a \rightarrow s_b$, i.e. it restricts the domain of the feasible action. Thus, given an N-state s_a for which A is a feasible action, A can be applied at s_a only if the N-state s_b that results from the application of A has certain compound features that are specified in $d(s_b)$.

Let $\{(A)\}$ be the (finite) set of rules of action and let $\{s\}$ be the set of all possible N-states. The set $\{(A)\}$ specifies a relation of *direct attainability* between the elements of $\{s\}$. Given any two states s_x, s_y from $\{s\}$, the N-state s_y is directly attainable from s_x if and only if there exists a permissible action in $\{(A)\}$ that can take s_x to s_y. Let us denote by T the relation of direct attainability.[1] The expression $s_x T s_y$ asserts that the N-state s_x can occur *just earlier than* s_y in a possible evolution of the system. Thus, the relation T represents local time order for the system P.

A *trajectory* from an N-state s_a to an N-state s_b is a finite sequence $s_1, s_2, \ldots, s_m$ of N-states such that $s_1 = s_a$, $s_m = s_b$, and for each i, $1 < i \leqslant m$, s_i is directly attainable from s_{i-1}. For any pair of N-states s_a, s_b, we say that *s_b is attainable from s_a* if and only if $s_a = s_b$ or there exists a trajectory from s_a to s_b. We denote the relation of attainability from s_a to s_b by $s_a \Rightarrow s_b$. The notion of a schedule is close to the notion of a trajectory; it is the sequence of actions that are taken in moving over the trajectory.

Now a problem of reasoning about actions can be formulated in the system P as follows: Given

(i) an N-state language
(ii) an extended description language
(iii) a set of rules of action
(iv) an initial N-state and a terminal N-state,

find the shortest schedule (or the shortest trajectory) from the initial N-state to the terminal N-state (if a schedule exists at all).

The set of all N-states, partly ordered under the relation T, defines a space σ that we call the N-state space. The search for a solution trajectory takes place in this space.

[1] This relation is very close to the relation 'earlier' introduced by Carnap (1958), and denoted T, in his language for space-time topology. In Carnap's case, T represents time order between two world points that are on the same trajectory.

Commonly, the initial formulation of a problem of reasoning about actions is a *verbal formulation*. Given the initial verbal formulation, there are several possible N-state languages and extended description languages that can be used for formulating the problem in the system of productions P. The choice of the universe U_1 and of the features in terms of which situations are described can strongly influence the amount of effort that is needed in order to find a solution in the formulation P. Here is an important decision point where problem solving power is affected by the choice of a problem representation. In addition, strong improvements in problem solving power may result from the discovery and exploitation of regularities in N-state space. The discovery of such regularities is facilitated by appropriate representations of N-state space. We shall illustrate these points by discussing in detail in the following sections a sequence of formulations of an extended version of the Missionary and Cannibals problem.

3. TRANSPORTATION PROBLEMS: INITIAL FORMULATION, F_1, OF M&C PROBLEMS

Many transportation scheduling problems are problems of reasoning about actions. Such problems can be formulated as follows. Given a set of space points, an initial distribution of objects in these points, and transportation facilities with given capacities; find an optimal sequence of transportations between the space points such that a terminal distribution of objects in these points can be attained without violating a set of given constraints on possible intermediate distribution of objects.

An interesting subclass of these transportation scheduling problems is the class of 'difficult crossing' problems, typified by the 'Missionaries and Cannibals' problem. This problem appears frequently in books on mathematical recreations. It has also received attention in the dynamic programming literature (Bellman and Dreyfus, 1962) and in the literature on computer simulation of cognitive processes. (Simon and Newell, 1961). The following is a verbal formulation of the 'missionaries and cannibals' problem (we call it formulation F_1). Three missionaries and three cannibals seek to cross a river (say from the left bank to the right bank). A boat is available which will hold two people, and which can be navigated by any combination of missionaries and cannibals involving one or two people. If the missionaries on either bank of the river, or 'en route' in the river, are outnumbered at any time by cannibals, the cannibals will indulge in their anthropophagic tendencies and do away with the missionaries. Find the simplest schedule of crossings that will permit all the missionaries and cannibals to cross the river safely.

In a more generalized version of this problem, there are N missionaries and N cannibals (where $N \geqslant 3$) and the boat has a capacity k (where $k \geqslant 2$). We call this problem the M&C *problem*. We shall refer to the specific problem that we have formulated above (where $N=3, k=2$) as *the elementary* M&C *problem*.

4. FORMULATION F_2 OF THE M&C PROBLEM IN ELEMENTARY SYSTEMS OF PRODUCTIONS

We shall formulate now the M&C problem in a system of productions of the type described in section 2. We start by specifying a simple but straightforward N-state language.

The universe U_0 of the N-state language contains the following basic elements:

(i) N individuals $m_1, m_2, \ldots, m_N$ that are missionaries and N individuals $c_1, c_2, \ldots, c_N$ that are cannibals,
(ii) an object (a transportation facility)–the boat b_k with a carrying capacity k,
(iii) two space points p_L, p_R for the left bank and the right bank of the river respectively.

The basic relations between basic elements in U_0 are as follows:

(i) *at*; this associates an individual or the boat with a space point (example: *at* (m_1, p_L) asserts that the missionary m_1 is at the left bank),
(ii) *on*; this indicates that an individual is aboard the boat (example: *on* (c_1, b_k) asserts that the cannibal c_1 is on the boat).

A set of expressions, one for each individual and one for the boat (they specify the positions of all the individuals and of the boat) provides a basic description of a situation, i.e. it characterizes an N-state. Thus, the initial N-state for the M&C problem can be written as follows:

$$s_0 = at(b_k, p_L), at(m_1, p_L), at(m_2, p_L), \ldots, at(m_N, p_L), at(c_1, p_L), at(c_2, p_L), \ldots, at(c_N, p_L). \tag{4.1}$$

The terminal N-state is attained from (4.1) by substituting p_R for p_L throughout.

The verbal statement of the M&C problem induces the formulation of an extended description language where a non-empty extension of U_0 is introduced together with certain properties and relations for the elements of this extension. The compound elements in the extension of U_0 are defined in terms of notions in the N-state language. These compound elements are the following six subsets of the total set $\{m\}$ of missionaries and the total set $\{c\}$ of cannibals:

$\{m\}_L = \{x \mid x \in \{m\}, at(x, p_L)\}$; the subset of missionaries at left,
$\{m\}_R = \{x \mid x \in \{m\}, at(x, p_R)\}$; the subset of missionaries at right,
$\{m\}_b = \{x \mid x \in \{m\}, on(x, b_k)\}$; the subset of missionaries aboard the boat.

The three remaining compound elements $\{c\}_L, \{c\}_R, \{c\}_b$ are subsets of the total set of cannibals that are defined in a similar manner.

In the M&C problem, the properties of interest for the specification of permissible actions are the sizes of the compound elements that we have just introduced, i.e. the number of elements in the subsets $\{m\}_L$, $\{m_R\}$, etc. Let M_L, M_R, M_b, C_L, C_R, C_b denote the number of individuals in the sets $\{m\}_L$, $\{m\}_R$, . . ., $\{c\}_b$ respectively. These are variables that take values from the finite set of nonnegative integers $J_0^N=\{0, 1, 2, \ldots, N\}$. These integers are also elements of the extension of U_0. They bring with them in the extended description language the arithmetic relations $=$, $>$, $<$, as well as compound relations that are obtainable from them via the logical connectives $\sim$, $\vee$, $\wedge$, and also the arithmetic functions $+$, $-$. A derived description $d(s)$ which is associated with an N-state s is a set of expressions that specify certain arithmetic relations between the variables M_L, M_R, etc. whose values are obtained from s.

The rules of formation that we shall use for description languages are of the type conventionally used in logic; they yield linear expressions. Expressions are concatenated (with separating commas) to form configurations. The basic description given in (4.1) is an example of a configuration in the linear language.

The verbal statement of the M & C problem does not induce a unique choice of a set of feasible actions. We shall consider first a 'reasonable' set of *elementary* actions that are assumed to be feasible and that satisfy the given constraints on boat capacity and on the possible mode of operating the boat. The set of permissible actions is a subset of this set that can be obtained by specifying the appropriate restrictions on the relative number of missionaries and cannibals in the two river banks as well as *'en route'*.

$\{(A)'\}_1$: *Elementary feasible actions in Formulation F_2 that are sensitive to boat constraints.* In the following transition schemata, α denotes an arbitrary configuration that completes a basic description of an N-state:

Load boat at left, one individual at a time $(LBL)'$
For any individual x,
$(LBL)'$: α, $at(b_k, p_L)$, $at(x, p_L)$; $(M_b+C_b\leqslant k-1)\rightarrow$ α, $at(b_k, p_L)$, $on(x, b_k)$; Λ
Move boat across the river from left to right $(MBLR)'$
$(MBLR)'$: α, $at(b_k, p_L)$; $(M_b+C_b>0)\rightarrow$ α, $at(b_k, p_R)$; Λ
Unload boat at right, one individual at a time $(UBR)'$.
For any individual x,
$(UBR)'$: α, $at(b_k, p_R)$, $on(x, b_k)$; $\Lambda\rightarrow$ α, $at(b_k, p_R)$ $at(x, p_R)$; Λ

In addition, we have the three following elementary actions in $\{(A)'\}_1$ 'Load boat at right one individual at a time $(LBR)'$', 'Move boat across the river from right to left $(MBRL)'$', and 'Unload boat at left one individual at a time $(UBL)'$'. The definitions of these actions are obtained from the previous definitions by substituting p_L for p_R and p_R for p_L in the corresponding actions. For example, the definition of $(MBRL)'$, is as follows:

$(MBRL)'$: α, $at(b_k, p_R)$; $(M_b+C_b>0)\rightarrow$ α, $at(b_k, p_L)$:

The six elementary actions that we have just introduced can be used together in certain sequences to form macro-actions for transfering *sets* of individuals from one river bank to the other. A transfer of r individuals from left to right, where $1\leqq r\leqq k$; can be effected by a sequence

$$\underbrace{(LBL)', (LBL)', \ldots, (LBL)'}_{r\text{ times}}, (MBLR)', \underbrace{(UBR)', (UBR)', \ldots, (UBR)'}_{r\text{ times}} \tag{4.2}$$

This sequence of actions starts with an empty boat at left and ends with an empty boat at right.

We can view the sequence of elementary actions in (4.2) as a transfer macroaction that is composed of two parts: the first part consists of the initial loading sequence for the boat, or equivalently the unloading sequence for the place that is the origin of the transfer. The second part starts with the river crossing and is followed by an unloading sequence for the boat, or equivalently by the loading sequence for the place that is the destination of the transfer. Since the constraints of the problem are given in terms of the relative sizes of various sets of individuals at points that can be considered as ends of loading (or unloading) sequences, then it is reasonable to attempt the formulation of actions as transitions between such points. We use these considerations in the formulation of a set of feasible compound actions that are only sensitive to boat constraints.

$\{(A)'\}_2$: *Compound feasible actions in formulation F_2 that are sensitive to boat constraints,*

Load empty boat at left with r individuals, $1\leqslant r\leqslant k$, ${}^r(L^rBL)'$.

Here we have a class of transition schemas that can be specified as follows:
For a set of r individuals $x_1, \ldots, x_r$, where $1\leqq r\leqq k$,

$(L^rBL)'$: α, $at(b_k, p_L)$, $at(x_1, p_L)$, . . ., $at(x_r, p_L)$; $(M_b+C_b=0)\rightarrow$ α, $at(b_k, p_L)$, $on(x_1, b_k)$, . . ., $on(x_r, b_k)$; Λ

In these transitions, r is the number of individuals from the left bank that board the boat for a crossing.

Move boat (loaded with r individuals) across the river from left to right and unload all its passengers at right $(MBLR+U^rBR)'$.

Here also we have a class of transition schemas which is defined as follows:
For a set of r individuals $x_1, \ldots, x_r$, $1\leqslant r\leqslant k$,
$(MBLR+U^rBR)'$: $\alpha[e]$, $at(b_k, p_L)$, $on(x_1\, b_k)$, . . ., $on(x_r, b_k)$; $\Lambda\rightarrow$ $\alpha[e]$, $at(b_k, p_R)$, $at(x_1, p_R)$, . . ., $at(x_r, p_R)$; Λ,

where $\alpha[e]$ stands for a configuration that is constrained by the condition e, which is as follows: no expression in the form $on(y, b_k)$, for any individual y is included in α. This is a way of saying that, after the crossing, all the r

passengers that have initially boarded the boat in the left bank, have to leave the boat and join the population of the right bank.

In addition to the two compound actions defined above, we have the two following compound actions in $\{(A)'\}_2$: 'Load empty boat at right with r individuals, $(L'BR)'$,' and 'Move boat (loaded with r individuals) across the river from right to left and unload all its passengers at left $(MBRL+U'BL)$''. The definitions of these compound actions are obtained from the definitions for $(L'BL)'$ and $(MBLR+U'BR)'$ by substituting p_L for p_R and p_R for p_L in the corresponding compound actions.

The compound actions that we have just introduced define the feasible transitions between N-states that are constrained only by the conditions on the transportation facility. Consider now a restriction on these compound actions that provides a set of rules of action where consideration is given to all the constraints of the M&C problem.

$\{(A)\}_2$: *First set of rules of action in formulation* F_2.

$(L'BL)$.

For a set of r individuals $x_1, \ldots, x_r$, where $1 \leqslant r \leqslant k$,

$$(L'BL)\colon \alpha,\ at(b_k, p_L), at(x_1, p_L), \ldots, at(x_r, p_L); (M_b+C_b=0) \rightarrow$$
$$\alpha,\ at(b_k, p_L), on(x_1, b_k), \ldots, on(x_r, b_k); ((M_L=0) \vee (M_L \geqslant C_L)),$$
$$((M_b=0)\ \vee (M_b \geqslant C_b)).$$

These compound actions are a subset of the compound actions $(L'BL)'$, where a valid next N-state is such that if any missionaries remain in the left bank then their number is no smaller than the number of cannibals remaining there, and also if any missionaries board the boat, then their number is no smaller than the number of cannibals that have also boarded the boat. Note that if an individual, say a missionary, is aboard the boat and the boat is at p_L, then the individual is not considered as a member of $\{m\}_L$, and therefore he is not counted in M_L.

$(MBLR+U'BR)$.

For any r, where $1 \leqslant r \leqslant k$,

$$(MBLR+U'BR)\colon \alpha[e],\ at(b_k, p_L), on(x_1, b_k), \ldots, on(x_r, b_k); \Lambda \rightarrow \alpha[e],$$
$$at(b_k, p_R), at(x_1, p_R), \ldots, at(x_r, p_R),$$
$$((M_R=0) \vee (M_R \geqslant C_R)).$$

Here the restricted configuration $\alpha[e]$ has the same meaning as in $(MBLR+U'BR)'$. The present compound actions are a subset of $(MBLR+U'BR)'$, where a valid next N-state is such that if any missionaries are present in the right bank then their number is no smaller than the number of cannibals there.

In addition to the transitions $(L'BL)$ and $(MBLR+U'BR)$, we also have the two transitions $(L'BR)$ and $(MBRL+U'BL)$, that are obtained from the previous ones by appropriately interchanging the places p_L and p_R throughout the definitions.

With the formulation of the permissible transitions between N-states, it is now possible to specify a procedure for finding a schedule of transfers that would solve the general M&C problem. Each transfer from left to right will be realized by a sequence $(L'BL)$, $(MBLR+U'BR)$, and each transfer from right to left will be realized by a sequence $(L'BR)$, $(MBRL+U'BL)$. Essentially, the selection of compound actions for each transfer amounts to finding r-tuples of individuals from a river bank that could be transferred to the opposite bank in such a way that cannibalism can be avoided in the source bank, in the destination bank and in the boat; i.e. the ***non-cannibalism conditions***

$$((M_L=0)\ \vee\ (M_L \geqslant C_L)), ((M_b=0)\ \vee\ (M_b \geqslant C_b)), ((M_R=0)\ \vee\ (M_R \geqslant C_R)) \qquad (4.3)$$

are all satisfied at the end of each of the two compound actions that make a transfer.

The formulation of compound actions and of problem solving procedures can be simplified *via* the utilization of the following property of our problem: *Theorem.* If at both the beginning and the end of a transfer the non-cannibalism conditions $((M_L=0)\ \vee\ (M_L \geqslant C_L))$ and $((M_R=0)\ \vee\ (M_R \geqslant C_R))$ are satisfied for the two river banks, then the non-cannibalism condition for the boat, i.e. $((M_b=0)\ \vee\ (M_b \geqslant C_b))$, is also satisfied.

Proof. At the beginning and the end of each transfer we have $M_L+M_R=C_L+C_R=N$; also, by supposition, the following two conditions hold simultaneously both at the beginning and at the end of a transfer:

$$(1)\ ((M_L=0)\ \vee\ (M_L=C_L)\ \vee\ (M_L>C_L)),$$
$$(2)\ ((N-M_L=0)\ \vee\ (N-M_L=N-C_L)\ \vee\ (N-M_L>N-C_L)). \qquad (4.4)$$

The conjunction of the above two conditions is equivalent to the following condition:

$$(M_L=0)\ \vee\ (M_L=N)\ \vee\ (M_L=C_L). \qquad (4.5)$$

But now in order to maintain this condition over a transfer, the boat can either carry a pure load of cannibals (to conserve $(M_L=0)$ or $(M_L=N)$) or a load with an equal number of missionaries and cannibals (to conserve $(M_L=C_L)$) or a load with a number of missionaries that exceeds the number of cannibals (for a transition from $(M_L=N)$ to $(M_L=C_L)$ or $(M_L=0)$, or a transition from $(M_L=C_L)$ to $(M_L=0)$). This conclusion is equivalent to asserting the non-cannibalism condition for the boat, i.e. $((M_b=0)\ \vee\ (M_b \geqslant C_b))$.

The previous theorem enables us to eliminate the non-cannibalism condition for the boat when we formulate permissible actions for realizing a transfer from one side of the river to the other. This permits the introduction of a single compound action per transfer. We can write then a new set of rules of action as follows:

$\{(A)\}_3$: *Second set of rules of action in formulation F_2*

Transfer safely a set of r individuals from left to right ($T'LR$).

For a set of r individuals $x_1, \ldots, x_r$, where $1 \leqslant r \leqslant k$,

$$(T'LR)\colon \alpha,\ at(b_k, p_L),\ at(x_1, p_L), \ldots, at(x_r, p_L);\ (M_b + C_b = 0) \rightarrow$$
$$\alpha,\ at(b_k, p_R),\ at(x_1, p_R), \ldots, at(x_r, p_R);\ (M_b + C_b = 0),$$
$$((M_L = 0) \vee (M_L \geqslant C_L)),\ ((M_R = 0) \vee (M_R \geqslant C_R))$$

Transfer safely a set of r individuals from right to left ($T'RL$).

The definition of this transfer action is obtained from ($T'LR$) by interchanging the places p_L and p_R throughout the definition.

It is clear that the formulation of the second set of rules of action has the effect of appreciably reducing the size of the N-state space that has to be searched, relative to the search space for the first set of rules of action. The transfers act as macro-actions, on basis of which the solution can be constructed without having to consider the fine structure of their component actions (loading the boat, unloading, crossing the river), thus without having to construct and consider intermediate N-states that are not needed for the key decisions that lead to the desired schedule.

Note that the reduction of the search space becomes possible because of the use of a formal property of our problem that enables the elimination of a redundant condition. The examination of the set of conditions of a problem, with the objective of identifying eliminable conditions and of reformulating accordingly the N-state space over which search proceeds, is one of the important approaches towards an increase in problem solving power.

5. FORMULATION F_3 OF THE M&C PROBLEM IN AN IMPROVED SYSTEM OF PRODUCTIONS

The notions that we have initially introduced in the description languages of the production systems of the previous sections reflect a general *a priori* approach to problems of reasoning about actions (i.e. consider as basic elements the individuals, the objects and the places that are specified in the problem, and consider as basic relations the elementary associations of individuals to places, etc), and also a problem–specific process of formulating concepts and attributes that are suggested from the verbal statement of the problem and that appear necessary for the expression of permissible transitions in the N-state space (notions such as M_L, C_L, etc. and the associated integers and arithmetic relations).

After several formulations of the problem, it becomes apparent that the description languages can be *restricted* and the formulation of N-states and of transitions between N-states can be considerably simplified. First, it is obvious that there is no need to use distinct individuals in the formulations. It suffices to use the compound elements, i.e. the sets $\{m\}_L$, $\{m\}_R$, $\{m\}_b$, $\{c\}_L$, $\{c\}_R$, $\{c\}_b$. Furthermore, since the conditions of the problem are expressed as arithmetic properties of the sizes of the compound elements, it suffices to consider the entities M_L, M_R, M_b, C_L, C_R, C_b, the set of integers $J_0{}^n$ and the arithmetic relations and operations. The main idea in this language restriction is that only those elements are to remain that are necessary for expressing the rules of action–that define the permissible transitions between N-states.

Because of the conservation of the total number of missionaries and the total number of cannibals throughout the transportation process, we have for each N-state (i.e. for each beginning and end of a transfer action) the following relationships:

$$M_L + M_R = C_L + C_R = N. \tag{5.1}$$

Thus, it is sufficient to consider explicitly either the set M_L, M_b, C_L, C_b or the set M_R, M_b, C_R, C_b; we choose to consider the former. Finally, we introduce two variables B_L, B_R in the restricted language such that

$$at(b_k, p_L) \equiv (B_L = 1) \equiv (B_R = 0)$$
$$at\ (b_k, p_R) \equiv (B_L = 0) \equiv (B_R = 1). \tag{5.2}$$

In the restricted N-state language the basic description of an N-state has the form

$$(M_L = i_1),\ (C_L = i_2),\ (B_L = i_3),$$

where i_1, i_2 are integers from J_0^N, and i_3 is 1 or 0. Such a description can be abbreviated to take the form of a vector (M_L, C_L, B_L), whose components are the numerical values of the key variables. The vector description shows explicitly the situation at the left river bank. Thus, the initial N-state of the M&C problem–expressed in the abbreviated vector notation–is (N,N,1), and the terminal N-state is (0,0,0).

We can now express the rules of action as follows:

$\{(A)\}_4$: *Set of rules of action in Formulation F_3.*

Transfer safely a mix (M_b, C_b) from left to right (TLR, M_b, C_b).

Any pair (M_b, C_b) such that $1 \leqslant M_b + C_b \leqslant k$, specifies a feasible action; for each such pair, we have a transition:

$$(TLR, M_b, C_b)\colon (M_L, C_L, 1);\ \Lambda \rightarrow (M_L - M_b, C_L - C_b, 0);$$
$$((M_L - M_b = 0) \vee (M_L - M_b \geqslant C_L - C_b)),$$
$$((N - (M_L - M_b) = 0) \vee (N - (M_L - M_b) \geqslant N - (C_L - C_b))).$$

Here M_b, C_b are the number of missionaries and the number of cannibals respectively that are involved in the transfer.

Transfer safely a mix (M_b, C_b) from right to left (TRL, M_b, C_b).

Again, any pair (M_b, C_b) such that $1 \leqslant M_b + C_b \leqslant k$, specifies a feasible action; for each such pair, we have a transition:

$$(TRL, M_b, C_b)\colon (M_L, C_L, 0);\ \Lambda \rightarrow (M_L + M_b, C_L + C_b, 1);$$
$$((M_L + M_b = 0) \vee (M_L + M_b \geqslant C_L + C_b)),$$
$$((N - (M_L + M_b) = 0) \vee (N - (M_L + M_b) \geqslant N - (C_L + C_b))).$$

The restriction of the N-state language, and the introduction of new basic descriptions for N-states and of new rules of transitions between N-states has a significant effect on the relative ease with which a solution of the M&C problem can be found. The irrelevant variety of transitions that is possible when individuals are considered, is now reduced to a meaningful variety that depends on the relative sizes of appropriately defined groups of individuals. In reasoning about the M&C problem, a completely different viewpoint can now be used. We do not have to think of individuals that are being run through a sequence of processes of loading the boat, moving the boat, etc. but we can concentrate on a sequence of vector additions and subtractions that obey certain special conditions and that should transform a given initial vector to a given terminal vector. The construction of a solution amounts to finding such a sequence of vector operations. The transition to the present formulation of the M&C problem illustrates an important process of improving a problem solving system by choosing an 'appropriate' N-state language and by using this language in an 'appropriate' way to define N-states and transitions between them.

6. FORMULATION F_4 OF THE M&C PROBLEM IN A REDUCTION SYSTEM

The previous formulations F_2 and F_3 of the M&C problem were in systems of productions. A solution to our problem in these systems amounts to finding the shortest schedule (or the shortest trajectory) from the intial N-state to the terminal N-state, if there exists a trajectory between these states (i.e. if there exists a solution at all). Note that this is a typical problem of derivation.

Let us formulate now the problem in a form that will permit us to specify a reduction procedure[1] for its solution. To specify the search space for the reduction procedure we need the notions of problem states (P-states) and the set of relevant moves – terminal and nonterminal. These notions correspond respectively to formulas, axioms and rules of inference in some natural inference system (Amarel, 1967).

P-states are expressions of the form $S=(s_a \Rightarrow s_b)$. In its logic interpretation, such an expression is a proposition that means 's_b is attainable from s_a'. Thus, it is equivalent to the logical notion CAN (s_a, s_b) that has been used by McCarthy (1963) and Black (1964) (in their formalization of problems of 'ordinary reasoning'), and that has been recently discussed by Newell (1966) and Simon (1966).

In the following, we consider the formulation F_3 in the improved system of productions as the starting point for the present formulation F_4. Thus, the initial P-state for the general M&C problem is

$$S_0=((N,N,1)\Rightarrow(0,0,0)). \tag{6.1}$$

A relevant *nonterminal move* corresponds to the application of a permissible action at the left N-state of a P-state. Thus, given a P-state $S_i=(s_a \Rightarrow s_b)$, and a permissible action A that takes s_a to s_c, then the application of the action at s_a corresponds to the application of a move (call it A also) that reduces S_i to the P-state $S_j=(s_c \Rightarrow s_b)$. We can represent such a move application as follows:

$$\begin{array}{l} S_i=(s_a \Rightarrow s_b) \\ \quad \Updownarrow A \text{ (a permissible action that takes } s_a \text{ to } s_c) \\ S_j=(s_c \Rightarrow s_b) \end{array}$$

In the logic interpretation, such a move corresponds to the inference 'S_j implies S_i' (this is the reason for the direction of the arrows). In other words, 'if s_b is attainable from s_c, then s_b is also attainable from s_a (because s_c is known to be attainable from s_a)'.

A terminal move in the present formulation, is a move that recognizes that the left and right sides of a P-state are identical; we call it M_t. Logically, such a move corresponds to the application of an axiom scheme for validation in the natural inference system.

A solution is a sequence of P-states, attained by successive applications of nonterminal moves, starting from the initial state and ending in a state where the terminal move applies. In the logic interpretation, a solution is a proof that the initial P-state is valid, i.e. that the terminal N-state is attainable from the initial N-state. From a solution in the reduction system, it is straightforward to attain a trajectory in the system of productions or the schedule of actions that is associated with such a trajectory.

7. THE SEARCH FOR SOLUTION IN THE REDUCTION SYSTEM

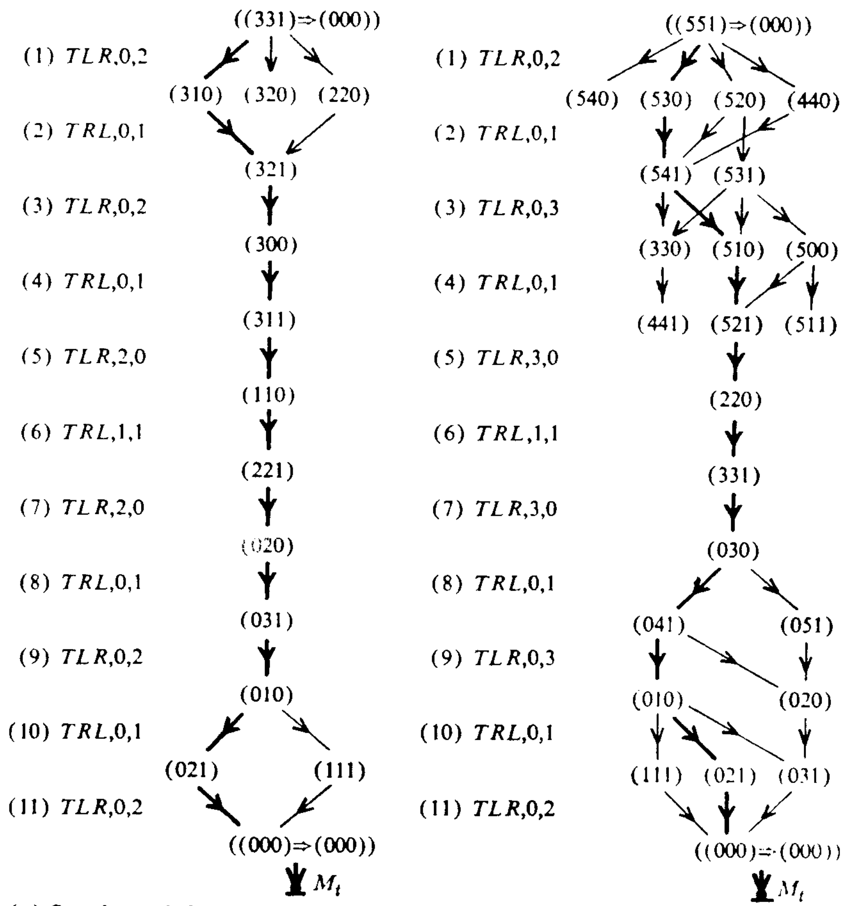
A simple search process by successive reductions can be used to obtain the solution. All relevant nonterminal moves are taken from a P-state. If a new P-state is obtained which is identical to a parent P-state in the search tree, then the development below that P-state stops. This guarantees the attainment of a simplest schedule if one exists and it provides a basis for a decision procedure, i.e. if all possible lines of development from the initial P-state are stopped, then no solution exists.

The search graphs for the cases ($N=3$, $k=2$) and ($N=5$, $k=3$) are shown in figure 7.1. These are condensations of search trees that are obtained by retaining only one copy of a P-state and its continuations. For simplicity, except for the initial and terminal P-states, all the P-states are represented by their left N-states (they all share the same right side; i.e. the desired terminal

[1] We have studied previously reduction procedures in the context of theorem-proving problems (Amarel, 1967) and syntactic analysis problems (Amarel, 1965). In these cases, the initial formulation of the problem was assumed to be in a system of productions. However, in the M&C problem, a formulation in a system of productions is a derived formulation that results from the translation of an initial verbal formulation.

N-state). The branches of the graphs represent move applications. The arrows indicate the direction of transfer actions for move applications. A solution is indicated in figure 7.1   a path in heavy lines. The schedule associated with a solution path is shown at the left of each graph as a sequence of transfer actions. Thus one (of the four possible) optional schedules for the elementary M&C problem ($N=3$, $k=2$) reads as follows:

(1) Transfer two cannibals from left to right.
(2) Transfer back one cannibal to the left.
⋮
(6) Transfer one missionary and one cannibal from right to left.
⋮
(11) Transfer two cannibals from left to right.



(a) Search graph for M&C problem with $N=3$, $k=2$

(b) Search graph for M&C problem with $N=5$, $k=3$

Figure 7.1. Search graphs for M & C problems in formulation F_4

In each case shown in figure 7.1 there is more than one solution. However, it is interesting to note that even if there is a certain amount of variety at the ends of the solution paths, the central part of the path has no variety (in the cases presented here, the center of the path is unique, in some other cases there may be two alternatives at the graph's neck, as we shall see in a subsequent example for $N=4$, $k=3$).

It should be evident from these search graphs that the M&C problem is a relatively simple problem that can be easily handled in an exhaustive search with a procedure of reduction type. There is no need for heuristics and complex rules for selecting moves and organizing the search. It is noteworthy that such a problem, while easily handled by computer procedures, is a relatively difficult problem for people. If one's approach is to try alternative sequences in some systematic manner (the computer approach that was just described) he becomes quickly memory limited. Also, people tend not to consider moves that, even though applicable to a situation, appear to be *a priori* bad moves on basis of some gross criterion of progress. In the elementary M&C problem, the sixth move in the schedule is such a stumbling block–yet it is the only move applicable.

Because of the one-sided development of the solution (from the initial N-state forward in time), and because of the exhaustiveness of the search, the process of searching for a solution would be the same if a reduction procedure (as described here) or a *generation procedure*, based directly on the formulation F_3, were used. In a generation procedure, all the sequences of N-states that are attainable from the initial N-state are constructed. The system is actually made to run over its permissible trajectories. The reduction approach was introduced at this stage, in order to show the equivalence between the generational approach (where the system is made to run between two given points) and the reductionist-logical approach (where essentially a proof is constructed that a trajectory exists between the two given points). While the reduction-logical approach has no advantage over the generational approach in the present formulation, there are cases where such an approach is especially useful. For example, in the next stage of formulation of the M&C problem it is convenient and quite natural to develop the approach to solution *via* a reduction procedure and its associated logical interpretation.

8. DISCOVERY AND UTILIZATION OF SYMMETRIES IN THE SEARCH SPACE. FORMULATION F_5 OF THE M&C PROBLEM

From an analysis of the search graphs for M&C problems (such as those in figure 7.1), it becomes apparent that the situation in search space is *symmetric with respect to time reversal*. Roughly, if we run a movie of a schedule of transportations forwards or backwards, we can't tell the difference. Consider two N-states (M_L,C_L,B_L) and $(N\text{-}M_L,N\text{-}C_L,1\text{-}B_L)$ in N-state space. When the space is viewed from the vantage point of each N-state in this pair, it appears identical, provided that the direction of transitions is 'perceived' by one N-

state as opposite to the direction 'perceived' by the other N-state. For example, consider the points (311) and (020) in the elementary M & C problem (see figure 7.1(a)). If we consider (311) on a normal time path, then it is reached *via* (TRL,0,1) and it goes to the next state *via* (TLR,2,0); if we consider (020) under time reversal, then it is reached *via* (TRL,0,1) and it goes to the 'next' state *via* (TLR,2,0). We shall consider now this situation more formally.

In our previous formulations of the M & C problem within production systems, the rules of action define a relation of direct attainability T between successive N-states (see section 2). Thus, for any two N-states s_a,s_b, the expression $s_a T s_b$ asserts that the N-state s_a occurs just earlier than s_b on a trajectory in N-state space. Consider now the converse relation $\check{T}$. The expression $s_a \check{T} s_b$ asserts that s_a occurs *just after* s_b on a trajectory.

We shall consider specifically in the following discussion the formulation of the M & C problem in the improved system of productions, i.e., the formulation F_8. Let σ be the space of N-states, partly ordered under the relation T, and $\check{\sigma}$ its dual space (i.e., $\check{\sigma}$ has the same elements of σ, partly ordered under $\check{T}$). Consider now the following mapping θ between N-states:

$$\theta\colon (M_L,C_L,B_L) \to (N\text{-}M_L,N\text{-}C_L,1\text{-}B_L) \tag{8.1}$$

We can also write θ as a vector subtraction operation as follows:

$$\theta(s)=(N,N,1)-s. \tag{8.2}$$

Theorem. For any pair of N-states s_a, s_b the following equivalence holds:

$$s_a T s_b \equiv \theta(s_a)\ \check{T}\ \theta(s_b),$$

or equivalently

$$s_a T s_b \equiv \theta(s_b)\ T\ \theta(s_a);$$

i.e. the spaces σ, $\check{\sigma}$ are anti-isomorphic under the mapping θ. Furthermore, the move that effects a permissible transition from s_a to s_b is identical with the move that effects a permissible transition from $\theta(s_b)$ to $\theta(s_a)$.

Proof. Consider any permissible N-state (i.e. the non-cannibalism conditions are satisfied at this state) with the boat at left; suppose that this N-state is described by the vector $s_a=(M_L, C_L, 1)$. Corresponding to s_a we have an N-state described by $\theta(s_a)=(N\text{-}M_L, N\text{-}C_L, 0)$. Note that, in general, the non-cannibalism conditions (stated in (4.4)) are invariant under θ. Thus, the N-state described by $\theta(s_a)$ is also permissible. We can also write in vector notation,

$$\theta(s_a)=(N,N,1)-s_a. \tag{8.3}$$

Consider now a transition from left to right at s_a, defined by some pair (M_b, C_b) such that $1 \leqslant M_b+C_b \leqslant k$. A transition of ths type is always *a priori* possible if $M_L+C_L \neq 0$ in s_a (i.e. if there is *somebody* at left when the boat is there–a condition which we are obviously assuming); however the *a priori* possible transition is not necessarily permissible–in the sense of satisfying the non-cannibalism conditions at the resulting N-state. The transition defined by (M_b, C_b) yields a new vector s_b that is related to s_a by vector subtraction as follows:

$$s_b=s_a-(M_b,C_b,1). \tag{8.4}$$

This can be verified by examining the rules of action. Corresponding to s_b we have via the mapping θ,

$$\begin{aligned}\theta(s_b)&=(N,N,1)-s_b=(N,N,1)-s_a+(M_b,C_b,1)\\ &=\theta(s_a)+(M_b,C_b,1).\end{aligned} \tag{8.5}$$

Suppose first that s_b is permissible (which means that the move defined by the pair (M_b, C_b) is permissible, and the relation $s_a\ T\ s_b$ holds); then $\theta(s_b)$ is also permissible because of the invariance of the non-cannibalism conditions under θ. Now in the N-state described by $\theta(s_b)$ the boat is at left and a left to right transition defined by (M_b, C_b) is possible (in view of (8.5) and noting that the components of $\theta(s_a)$ cannot be negative). This transition yields a vector $\theta(s_b)-(M_b, C_b, 1)$, which is identical with $\theta(s_a)$. Since $\theta(s_a)$ is permissible, then the transition defined by (M_b, C_b) (which takes $\theta(s_b)$ to $\theta(s_a)$) is permissible, and the relation $\theta(s_b)\ T\ \theta(s_a)$ holds. It is inherent in this argument that the same move that takes s_a to s_b, also takes $\theta(s_b)$ to $\theta(s_a)$.

Suppose now that s_b is not permissible (which means that the relation $s_a\ T\ s_b$ does not hold); then $\theta(s_b)$ is not permissible either, and the relation $\theta(s_b)\ T\ \theta(s_a)$ does not hold.

A similar argument can be developed for a right to left transition. This establishes the anti-isomorphism and the relationship between symmetric moves.

The situation can be represented diagramatically as follows:

$$\begin{tikzcd} s_a \arrow[r, dashed, "\theta"] \arrow[d, "T"'] & \theta(s_a) \arrow[d, shift left, "\check{T}"] \\ s_b \arrow[r, dashed, "\theta"] & \theta(s_b) \arrow[u, shift left, "T"] \end{tikzcd} \tag{8.6}$$

Corollary. For any pair of N-states s_a,s_b, the following equivalence holds:

$$(s_a \Rightarrow s_b) \equiv (\theta(s_b) \Rightarrow \theta(s_a)).$$

The proof is an extension of the previous proof.

The recognition of the anti-isomorphism permits us to approach the problem simultaneously, and in a relatively simple manner, both in the space σ and in its dual space. The reasoning behind this dual approach relies on the logical properties of the attainability relation $\Rightarrow$, and on the properties of the anti-isomorphism.

Consider an attainability relation $(s_0 \Rightarrow s_b)$, where s_0 is the initial N-state and s_b is an arbitrary N-state such that $s_b \neq s_0$. Let us denote by $\{s_1\}$ the set of all N-states that are directly attainable from s_0; thus

$$\{s_1\} = \{s | s_0 T s \text{ holds}\}. \tag{8.8}$$

We have then

$$(s_0 \Rightarrow s_b) \equiv \bigvee_{s \in \{s_1\}} (s \Rightarrow s_b). \tag{8.9}$$

If $s_b = s_t$, where s_t is the desired terminal N-state, then we have as a special case of (8.9),

$$(s_0 \Rightarrow s_t) \equiv \bigvee_{s \in \{s_1\}} (s \Rightarrow s_t). \tag{8.10}$$

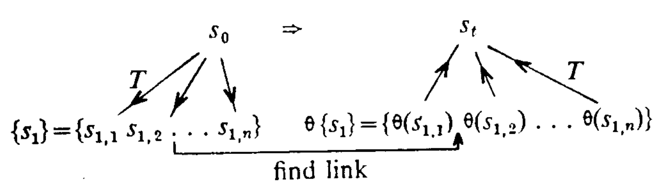
From the previous corollary, and since $\theta(s_t) = s_0$ in the M&C problem, we can write the equivalence (8.10) as follows:

$$(s_0 \Rightarrow s_t) \equiv \bigvee_{s \in \{s_1\}} (s_0 \Rightarrow \theta(s)). \tag{8.11}$$

By using (8.9) in (8.11) we obtain:

$$(s_0 \Rightarrow s_t) \equiv \bigvee_{s_i \in \{s_1\}} \Big(\bigvee_{s_j \in \{s_1\}} (s_j \Rightarrow \theta(s_i)) \Big). \tag{8.12}$$

The situation can be shown schematically as follows:



(8.13)

The terminal N-state s_t is attainable from s_0 if and only if any of the N-states from which s_t is directly attainable is itself attainable from any N-state that is directly attainable from s_0.

Now for each growth below $s_{1,i} \in \{s_1\}$, there is a corresponding *image growth* below $\theta(s_{1,i})$. Let us denote the set of all N-states that are directly attainable from elements of $\{s_1\}$ by $\{s_2\}$; thus

$$\{s_2\} = \{s | s_a \in \{s_1\},\ s_a T s \text{ holds}\}. \tag{8.14}$$

Let us call the image of $\{s_2\}$ under θ, $\theta\{s_2\}$. Repeating the previous argument we obtain that s_t is attainable from s_0 if and only if any of the N-states in $\theta\{s_2\}$ is attainable from any of the N-states in [illegible] This type of argument can be continued until either a set $\{s_n\}$ at som[illegible] n does not have any new progeny, or an N-state in $\theta\{s_n\}$ is directly a[illegible] from an N-state in $\{s_n\}$.

From the preceding discussion, it is clear that we can develop the search for solution simultaneously, both forward from the initial N-state and backward from the terminal N-state, without having to spend search effort in both sides. Only the sets $\{s_1\}, \{s_2\}, \ldots \{s_n\}$, that represent the forward exploration of the search space from the initial N-state, have to be constructed. The exploration from the terminal N-state backwards is directly obtainable as the image of the forward exploration under time reversal (i.e. under the anti-isomorphism). This means that the knowledge of the symmetry property permits us to cut the depth of search by a factor of two – which is a substantial reduction in expected search effort. Note, however, that as is the case in any two-sided approach to search, new problems of coordination and recognition arise because of the need to find links between the forward moving search front and its backward moving image. In our present problem, because of the relative narrowness of the moving fronts, this problem of recognizing a linking possibility is not too difficult.

Let us formulate now a reduction procedure for carrying out the two-sided solution construction activity that we have just described. We introduce here a broader concept of a problem state, the *total P-state*, Σ:

$$\Sigma_i = (\{s_i\} \Rightarrow \theta\{s_i\}),\ i = 0, 1, 2, \ldots$$

where i indicates the number of transitions from one of the schedule terminals (initial or terminal N-state) and the current total P-state. In its logic interpretation, an expression Σ_i stands for the proposition 'there exists an N-state in $\{s_i\}$ from which some N-state in $\theta\{s_i\}$ is attainable'.

A *nonterminal move* in the present formulation is a broader notion than a nonterminal move in our previous reduction procedure. Here, a nonterminal move effects a transition between Σ_i and Σ_{i+1} in such a manner that $\Sigma_i \equiv \Sigma_{i+1}$. Such a move represents a combination of parallel transfers, half of which are source-based and they are found by direct search, and the other half are destination-based and they are computed on basis of the symmetry property.

A *terminal move* in the present formulation establishes links between N-states in $\{s_i\}$ and N-states in $\theta\{s_i\}$ that are directly attainable from them.

A *solution* (or correspondingly an attainability proof) has the form of a chain of total P-states that start with $\Sigma_0 = (s_0 \Rightarrow s_t)$ and that ends with a total P-state Σ_n where a terminal move applies. A trajectory (or a schedule) is obtained from this solution by tracing a sequence of N-states that starts with s_0; it is followed by a directly attainable N-state in $\{s_1\}$; it continues this way up to $\{s_n\}$, and then it goes to $\theta\{s_n\}, \theta\{s_{n-1}\}, \ldots$ up to $\theta(s_0) = s_t$.

The development of the solution for the elementary M&C problem in the present formulation is shown in figure 8.1.

The total P-state Σ_5 is valid because there is a link (*via* TRL, 1,1) between 110 and 221. The darkened path shows a solution trajectory. The schedule associated with the trajectory is given at left. The same transfer actions apply at points of the trajectory that are equidistant from the terminals. Thus, in the

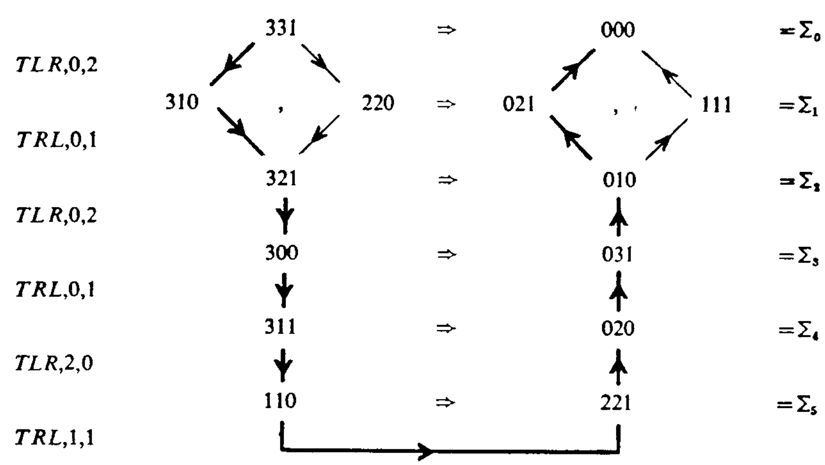


Figure 8.1. Search graph for the elementary M & C problem in the formulation F_6

present case, we have a schedule which is symmetrical with respect to its middle point. Note that the solution development given in figure 8.1 is a folded version of the solution development which is given in figure 7.1(a).

It is of interest to develop the solution for the case $N=4$, $k=3$ within the present formulation; this is given next in figure 8.2.

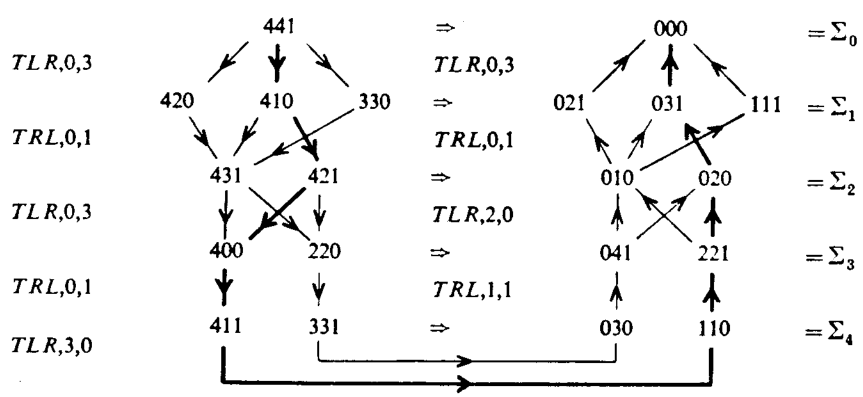


Figure 8.2. Search graph for the M & C problem ($N=4$, $k=3$) in the formulation F_6

The total P-state Σ_4 is valid, since a terminal move composed of two links applies at Σ_4. The darkened path in figure 8.2 shows one solution trajectory. The schedule associated with the trajectory is shown in the sides of the solution graph. Note that in the present case the trajectory is not symmetrical. While the two halves of the search graph are images of each other under θ, the two halves of a trajectory are not. Roughly the situation is as follows: Two main sequences of N-states grow from each of the two sides; these two sequences are images of each other under θ; a solution trajectory starts with one of these sequences from the one side, and then at its middle point, rather than continuing with the image of the initial sequence, it flips over to the image of the second sequence.

In the present formulation, it is possible again to devėlop a solution *via* a generation procedure that would operate in an equivalent manner to the reduction procedure that we have described here. However, the direct correspondence between the logic of the solution and the elements of the reduction procedure make the latter more convenient to use.

9. DISCOVERY OF SOLUTION PATTERNS IN AN APPROPRIATE REPRESENTATION OF N-STATE SPACE

One of the significant ways of increasing the power of a problem solving system for the M & C problem is to look for some characteristic patterns in its search space that go beyond the properties that we have discussed so far. To this end, it is extremely important to find a representation of the search space that enables a global view of the situation, so that reasoning about a solution can first proceed in broad terms and it can then be followed by the detailed scheduling of actions. We shall present next such a representation of the space of N-states. This representation utilizes the basic description of N-states that was introduced in the formulation F_3 of the M & C problem.

The number of possible N-states for an M & C problem equals the number of possible valuations of the vector (M_L, C_L, B_L); this number is $2(N+1)^2$. We represent the space of N-states by a limited fragment of three-dimensional space with coordinates M_L, C_L and B_L. This fragment consists of two parallel square arrays of points, that are disposed as follows: One array is on the plane $B_L=0$ and the other on the plane $B_L=1$; the points on each array have coordinates (M_L, C_L), where the values of M_L, C_L are 0, 1, 2, . . ., N. Thus, each point corresponds to a possible N-state. Such a representation for the N-state space of the elementary M & C problem is shown in figure 9.1. The blackened points stand for non-permissible N-states (i.e. the non-cannibalism conditions are violated in them). The feasible transitions from an N-state s in a given B_L plane to other N-states in the same plane are shown in figure 9.2. These feasible transitions reflect mainly boat capacity. A feasible transition is not permissible if it leads to a non-permissible N-state. Thus, starting from an N-state in the $B_L=1$ plane, a transition can be made to any permissible point within a 'distance' of 2 lattice steps in the plane, in a general southwestern direction; after the movement in the plane is carried out (it represents 'load-in the boat' at left) a left-to-right transfer action is completed by jumping from the $B_L=1$ array to the $B_L=0$ array in a direction parallel to the B_L axis. A right-to-left transfer starts from an N-state in the $B_L=0$ plane; a transition is first made to a permissible point within a 'distance' of 2 lattice steps in the plane, in a general northeastern direction; after this transition, the transfer is completed by jumping across to the $B_L=1$ array.

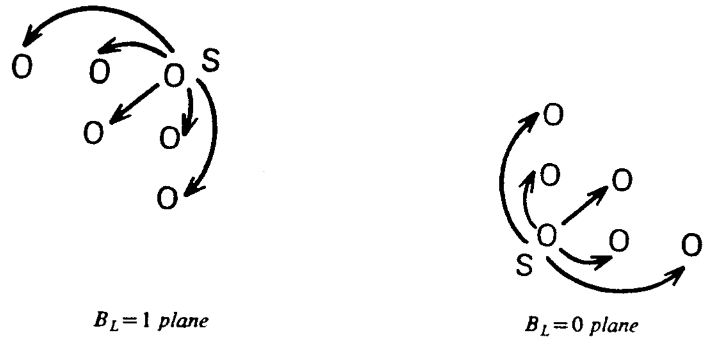


Figure 9.1 Feasible transitions in space of N-states

A solution for the elementary M & C problem is shown in figure 9.1 as a path in N-state space. It is suggestive to regard the solution path as a thread entering the initial N-state, leaving the terminal N-state, and woven in a specific pattern of loops that avoids going through the non-permissible points in N-space. Furthermore, the solution shown in figure 9.1 requires the 'least amount of thread' to go from the initial N-state to the terminal N-state within the imposed constraints in the weaving pattern. It is easy to see that the solution trajectory shown in figure 9.2 is the same as the solution shown in figure 7.1(a).

We can simplify the representation of N-state space by collapsing it into a single square array of $(N+1)^2$ points (figure 9.3). This requires a more complex specification of the possible transitions. We represent a left-to-right transfer by an arrow with a black arrowhead, and a right-to-left transfer by an arrow with a white arrowhead. In the previous two-array representation, a black arrow corresponds to a movement in the $B_L=1$ plane that is followed by a jump across planes, and a white arrow corresponds to a movement in the $B_L=0$ plane followed by a jump across planes. A point in the collapsed space is given by two coordinates (M_L,C_L), and it can represent either of the two N-states $(M_L, C_L, 1)$ or $(M_L, C_L, 0)$. The point (M_L, C_L) in association with an entering black arrowhead represents $(M_L,C_L,0)$; in association with an entering white arrowhead, it represents $(M_L,C_L,1)$. A sequence of two arrows → ⇾ represents a round trip left-right-left. A sequence of arrows, with alternating arrowhead types, that starts at the initial point (N,N) and ends at the terminal point $(0,0)$ represents a solution to the M & C problem.

The collapsed N-state space for the elementary M & C problem is shown in figure 9.3. The solution path shown in this figure represents the same solution

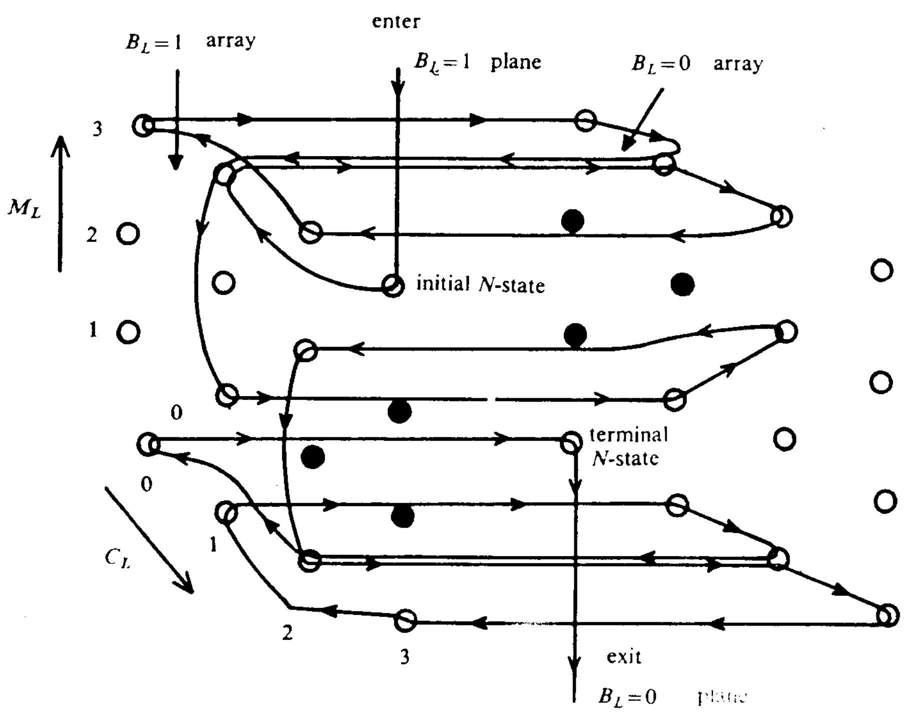


Figure 9.2. Space of N-states for [illegible] & C problem

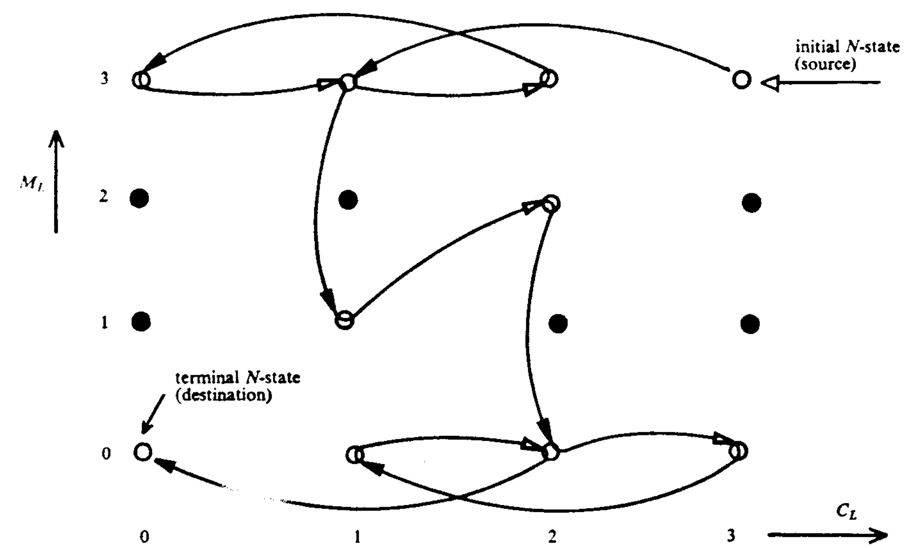


Figure 9.3. Collaps[illegible] pace for elementary M & C problem

that is shown (in different forms) in the figures 7.1 (a) and 9.2. The solution path in the collapsed N-state space suggests a general movement forward from the source point to the destination point by a sequence of 'dance steps' of the type 'two steps forward, one step back' over a dance floor made of white and black tiles, where black tiles are to be avoided (however, they can be skipped over).

It has been our experience that when the elementary M&C problem is presented to people in the form of pathfinding in the collapsed N-state space, the ease with which a solution is found is substantially higher than in any of the previous formulations. It appears that many significant features of the solution space are perceived simultaneously, attention focuses on the critical parts of the space, and most often the solution is constructed by reasoning first with global arguments and then filling in the detailed steps.

One of the features that are immediately noticed in examining the collapsed N-state space is that the 'permissible territory' for any M&C problem forms a **Z** pattern. The horizontal bars of the **Z** region correspond to the conditions $M_L=N$ and $M_L=0$, and the diagonal line corresponds to the condition $M_L=C_L$. The conditions that specify the 'permissible territory' can

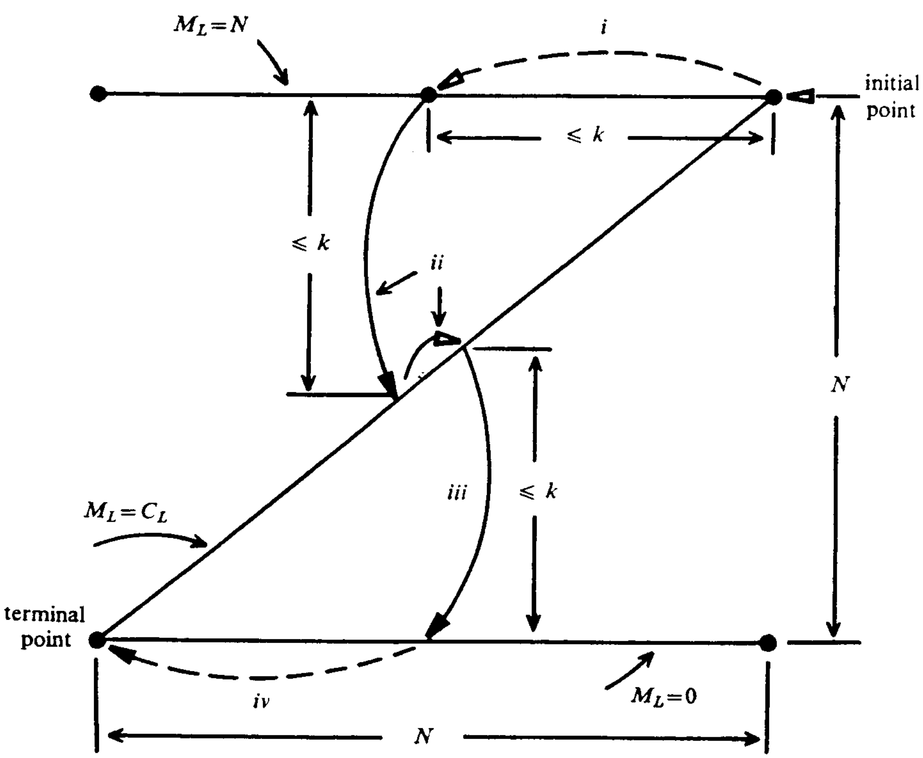


Figure 9.4. The 'permissible territory' in the M & C problem

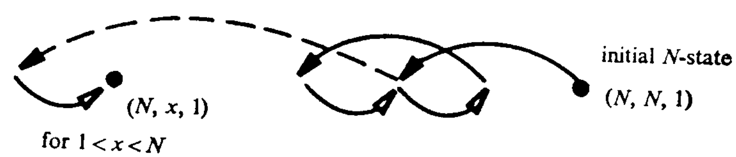
be obtained directly as consequences of the problem constraints; we have used them in the proof of the eliminability of the 'boat condition' in section 4, and it is conceivable that they could be derived mechanically with techniques that are presently available. Note, however, that the problem of obtaining these conditions is not a theorem proving task but a *theorem finding task*.

Let us concentrate now on the **Z** region of interest in the collapsed N-state space of an M&C problem, and let us attempt to find general characteristic features of solution paths. Since the **Z** region is the permissible territory, it is reasonable to expect that features of solution paths are describable in terms of movement types over this **Z**. By examining the diagram in figure 9·4 we shall try first to identify certain properties of solution paths that will permit us to characterize the solution schema that we have used in the elementary M&C problem (see figure 9.3).

In the diagram of the **Z** region, this solution schema can be seen to consist in general of four main parts, (i) to (iv). An arrow ◁– – – denotes a sequence of transitions the last of which brings the boat to the left river bank, and an arrow ◀– – – denotes a sequence of transitions that terminates with the boat at right.

The following general properties of solution paths are suggested by examining the situation in figure 9.4:

(i) On the $M_L=N$ line, any of the points $(N, x, 1)$, where $1<x<N$, are attainable from the initial point $(N,N,1)$ by a 'horizontal' sequence of transitions of the following type:



More generally, any point $(N, x, 1)$, where $1\leqslant x\leqslant N$, can be attained from any other point $(N, y, 1)$, where $1\leqslant y\leqslant N$, by some 'horizontal' sequence of transitions that is similar to the one just shown. Roughly, this indicates that 'horizontal' movements over the $M_L=N$ line are *easily achievable* by a known routine of steps.

(ii) If k is the boat capacity, and if $k\geqslant 2$, then any of the points $(N, N-x, 1)$, where $0<x\leqslant k$, can reach, *via* a single transition $(TLR, x, 0)$, a point $(N-x, N-x, 0)$ on the diagonal of the **Z** region. From this point, a $(TRL, 1, 1)$ transition can lead to a point $(N-x+1, N-x+1, 1)$ on the diagonal. While the first transition in this pair determines the size of the 'jump' from the $M_L=N$ line to the diagonal, the second transition is necessary for

'remaining' on the diagonal. Thus, we can regard this pair of transitions as a way of achieving a 'stable jump' from the line $M_L = N$ to the diagonal. It is clear from this discussion that a boat capacity of at least two is necessary for realizing a 'stable jump'. Note that the second transition in the pair corresponds to the critical move of returning one missionary and one cannibal–in general, an equal number of missionaries and cannibals–to the left, in mid schedule. As we have observed before, this is an unlikely move choice if the problem solver has a general notion of progress that guides his move preferences uniformly over all parts of the solution space. Only after knowing the local structure of this space, is it possible to see immediately the inevitability of this move. Now, the remotest point of the diagonal (from the initial point) that can be reached by this pair of transitions is $(N-k+1, N-k+1, 1)$.

(iii) A point on the diagonal can directly attain a point on the line $M_L=0$ if its distance from that line does not exceed k. Thus, to move from the $M_L=N$ line to the $M_L=0$ line in two 'jumps', by using the diagonal as an intermediate support, we need a boat capacity that satisfies the following condition:

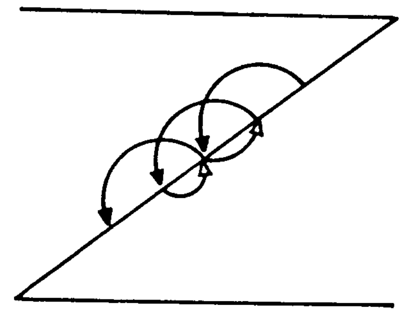
$$k \geq \frac{N+1}{2}. \tag{9.1}$$

(Thus, for $N=5$ and $k=2$ there is no solution. This specific result could have been obtained in any of our previous formulations by recognizing that a definite dead end is attained in the course of searching for a solution. However, it is obtained much more directly from our present analysis; furthermore, we can easily assign the reason for the unsolvability to the low capacity of the boat.)

(iv) On the $M_L=0$ line, any of the points to the right of the terminal point, can reach the terminal point (0,0,0) by a 'horizontal' sequence of transitions of the type shown in (i). More generally, any point $(0,x,0)$, where $0 \leq x < N$, can be attained from any other point $(0,y,0)$, where $0 \leq y < N$, by some 'horizontal' sequence of transitions. Again, this indicates roughly that 'horizontal' movement over the $M_L=0$ line are easily achieved by a known routine of steps.

From the general properties just discussed we can characterize a general solution pattern, which we call the *zig-zag pattern*, by the following sequence of global actions: (i) starting from the initial point, slide on the $M_L=N$ line, over a 'horizontal' transition sequence, up to the point $(N, N-k, 1)$; (ii) jump on the diagonal, *via* two transitions, to the point $(N-k+1, N-k+1, 1)$; (iii) jump off the diagonal to the M [illegible] line; (iv) slide on the $M_L=0$ line, *via* a 'horizonta[illegible]nsition sequenc[illegible] terminal point.

It can be easily verified th[illegible] solutions to [illegible] cases that [illegible] have presented previously, i.e. $(N=3, k=2)$, $(N=4, k=3)$ and $(N=5, k=3)$, follow precisely the zig-zag pattern that we have outlined. If $N=6$, then in order to use the present solution scheme, a boat of capacity 4 is needed (see the condition (9.1)). When a boat capacity of 4 (or more) is available, then *any* M & C problem is solvable. This property is due to the fact that the following pattern of transitions, that allows one 'to slide along the diagonal', is possible when $k \geq 4$:



The 'sliding along the diagonal' for $k=4$ is realized by a 'diagonal' sequence of round trips of the type: $(TLR, 2, 2), (TRL, 1, 1), (TLR, 2, 2), (TRL, 1, 1)$, etc., where each round trip realizes a net transfer of two individuals from left to right.

For cases with $k \geq 4$ it is possible to use a simple and efficient solution pattern, the *diagonal pattern*, that has a single global action, as follows: starting from the initial point slide down the diagonal via a 'diagonal' transition sequence that takes in each round trip $\frac{k}{2}$ missionaries and $\frac{k}{2}$ cannibals to the right (when k is even–otherwise it takes $\frac{k-1}{2}$ of each) and it returns one missionary and one cannibal back, except in the last trip, until the terminal point is reached. It is also possible to construct solution patterns that combine parts of the zig-zag pattern with parts of the diagonal pattern. Such a combined solution scheme is shown in figure 9.5.

For the M & C problem (i.e. find a path from $(N,N,1)$ to $(0,0,0)$), it can be shown that if the boat capacity k is high, and if k is even, then the pure diagonal pattern of solution is always better than any combined pattern (in terms of number of trips required for a schedule); if k is odd, then there are cases where a small advantage is gained by starting the schedule with the first two round trips of the zig-zag pattern; if $k=4$, and $N \geq 6$, then the diagonal solution pattern, the zig-zag pattern or the combined pattern of figure 9.5, when it applies, are all of equivalent quality.

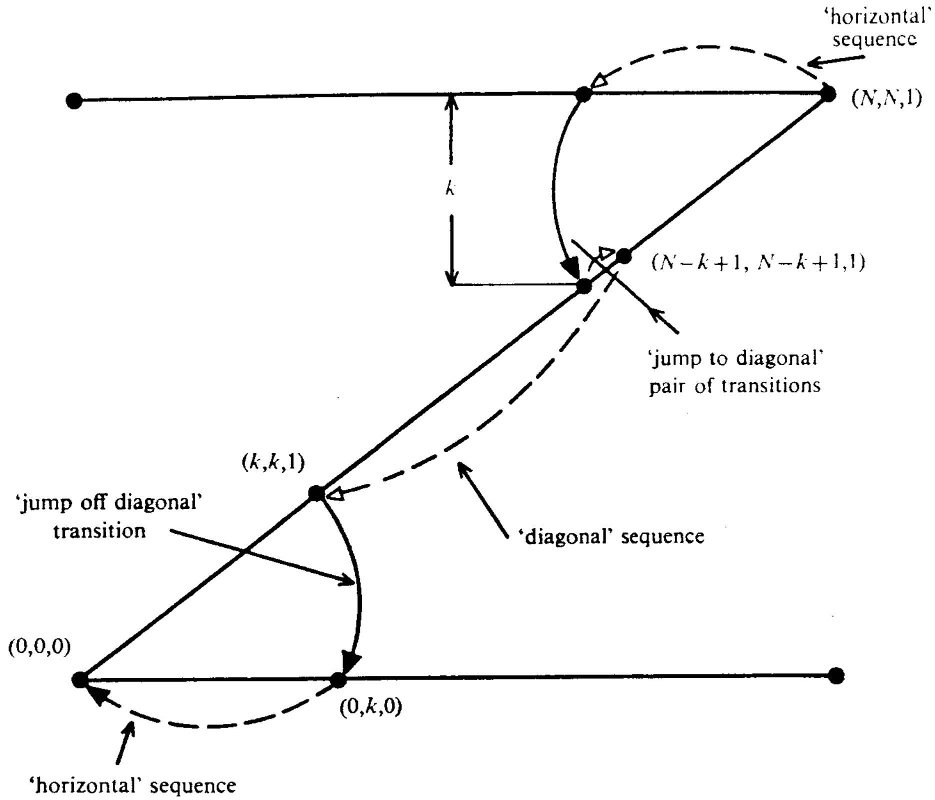


Figure 9.5. Combined scheme of solution shown on the Z region

10. FORMULATION F_6 OF EXTENDED M&C PROBLEM IN A MUCH IMPROVED PRODUCTION SYSTEM THAT CORRESPONDS TO A HIGHER LEVEL SEARCH SPACE

After the exploration of solution patterns in our array representation of N-state space, and after new global transition concepts are developed, it is possible to re-formulate the M&C problem (in fact, an extended version of this problem) in a new and much improved system of productions to which there corresponds an N-state space that has many fewer points than in any of the previous spaces.

From the analysis of possible global movements in the N-state space, we can now formulate the following set of *macro-transitions*:

$$
\begin{aligned}
&\{(A)\}_6\text{: set of rules of (macro) action in formulation } F_6.\\
&(H_1)\colon (N,C_L,1);\ 0<C_L<N,\ k\geqslant 2\rightarrow(N,N,1)\\
&(H_1,J_1)\colon (N,C_L,1);\ 0<C_L\leqslant N,\ k\geqslant 2\rightarrow(N-k+1,\ N-k+1,\ 1)\\
&(D)\colon (M_L,C_L,1);\ 0<M_L=C_L\leqslant N,\ k\geqslant 4\rightarrow(0,0,0)\\
&(J_2)\colon (M_L,C_L,1)\colon 0<M_L=C_L\leqslant k\rightarrow(0,C_L,0)\\
&(D,J_2)\colon (M_L,C_L,1);\ M=C_L>k\geqslant 4\rightarrow(0,k,0)\\
&(H_2)\colon (0,C_L,0);\ 0\leqslant C_L<N,\ k\geqslant 2\rightarrow(0,C_L',0);\ 0\leqslant C_L'<N,\ C_L\neq C_L'
\end{aligned}
\tag{10.1}
$$

Each of these macro-transitions is realized by a routine of elementary transitions. Thus, (H_1) is realized by a 'horizontal' sequence of transitions that slides a point on the $M_L=N$ line to the corner point $(N,N,1)$, with the least number of steps; (H_1,J_1) is realized by a 'horizontal' sequence of transitions that takes a point on the $M_L=N$ line to the point $(N,N-k,\ 1)$ on that line, and then it is followed by a pair of transitions that effects a 'stable jump' to the point $(N-k+1,\ N-k+1,\ 1)$ on the diagonal, all this with the least number of steps; (D) is realized by a 'diagonal' sequence of transitions that takes a point on the diagonal to the bottom of that diagonal, in the least number of steps; (J_2) is realized by a single transition that effects a 'jump' from a point on the diagonal to the $M_L=0$ line; (D,J_2) is realized by a 'diagonal' sequence of transitions that takes a point along the diagonal to the point $(k,k,1)$, and then it is followed by a transition that effects a 'jump' to the point $(0,k,0)$ on the $M_L=0$ line, all this with the least number of steps; (H_2) is realized by a 'horizontal' sequence of transitions that takes a point on the $M_L=0$ line to another point on that line, in the smallest number of steps.

The formulation of the macro-transitions enables us to approach a problem of finding the best schedule for an M&C problem (or extensions of this problem) by first solving the problem in a higher order space, where we obtain a set of possible *macro-schedules*–that are defined in terms of macro-transitions–and then converting the macro-schedules to schedules by compiling in the appropriate way the macro-transition routines. Note that the present formulation is suitable for handling conveniently a class of problems which is larger than the strict class of M&C problems that we have defined in section 3; specifically, an arbitrary distribution of cannibals at left and right can be specified for the initial and terminal N-states. By certain changes in the specification of the macro-transitions, it is possible to consider within our present framework other variations of the M&C problem, e.g. cases where the boat capacity depends on the state of evolution of the schedule, cases where a certain level of 'casualties' is permitted, etc.

Let us consider now the following example:

Example 10.1. The initial situation is as follows: nine missionaries and one cannibal are at the left river bank and eight cannibals are at the right bank; a boat that has a capacity of four is initially available at left. We wish to find the simplest safe schedule that will result in an interchange of populations between the two river banks.

The search graph in the higher order space gives all the macro-schedules for the case of a constant boat capacity of four; this graph is shown in figure 10.1. The macro-transitions are applied on the left side of a P-state (i.e., the macro-schedule is developed forward in time) until a conclusive P-state is reached. The number within square brackets that is associated with a macro-transition indicates its 'weight', i.e., the number of trips in the routine that realizes the macro-transition. Thus, we have macro-schedules of weights 15, 21, and 27. The simplest macro-schedule is given by the sequence

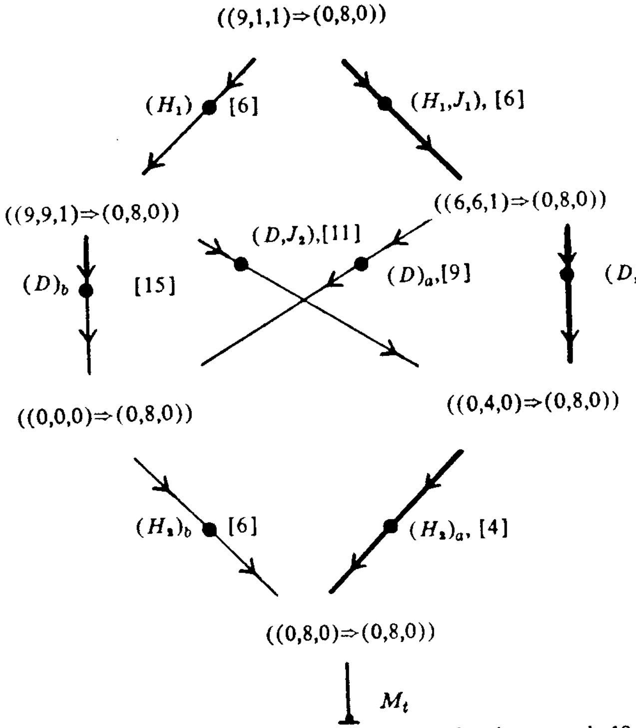


Figure 10.1 Search graph in higher order space for the example 10.1

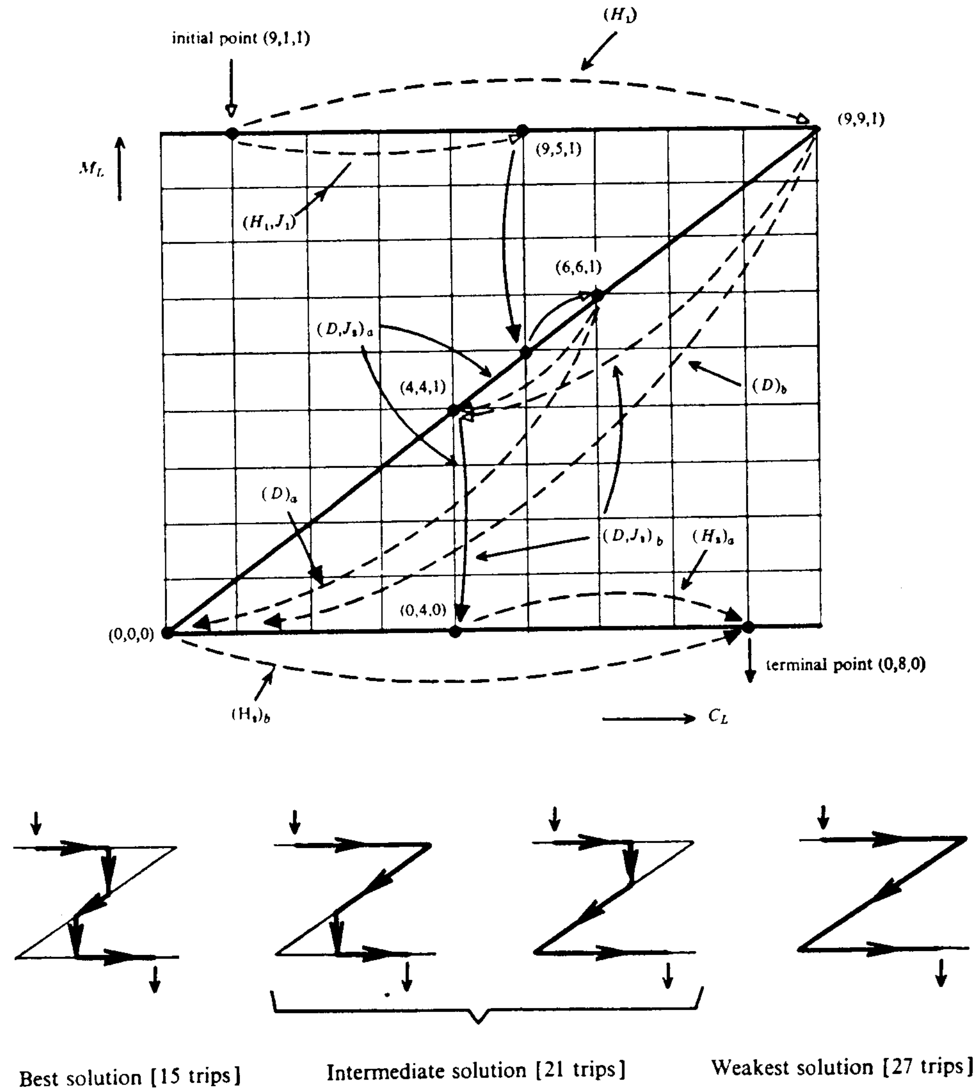
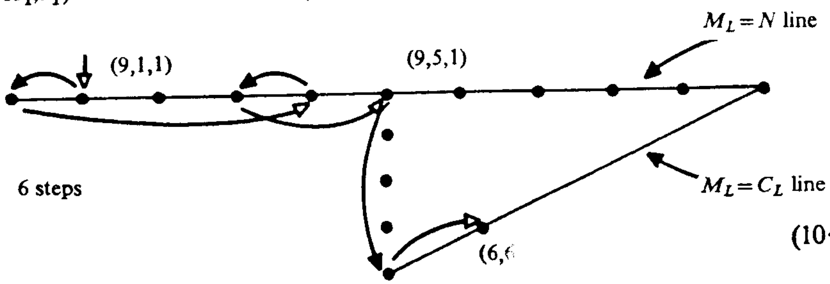


Figure 10.2. Collapse space for the example (10.1)

(H_1,J_1), $(D,J_2)_a$, $(H_2)_a$ of macro-transitions, which corresponds to the darkened path in figure 10.1.

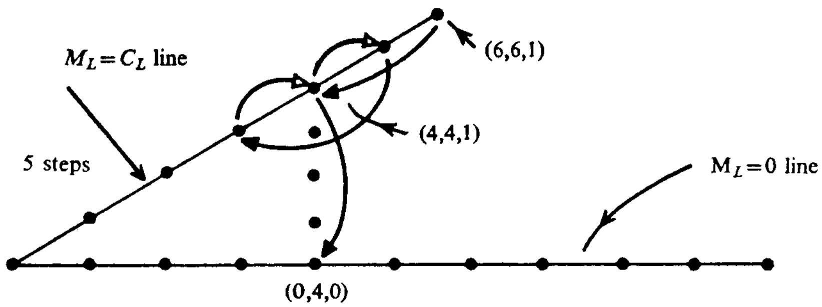
The situation in the collapsed N-state space is shown in figure 10.2. The patterns of the alternative macro-schedules are shown schematically in the lower part of the figure.

After a macro-transition is specified, its realization in terms of elementary transitions is easily carried out by a compiling routine. For example, the macro-transition (H_1,J_1) in our problem is realized as follows by a routine (H_1,J_1) with initial N-state (9,1,1) and a terminal N-state (6,6,1):



(10·2)

As a second example, consider next the real of the macro-transition $(D,J_2)_a$, by a routine (D,J_2) from (6.6.1) to 0); *see* (10.3).



(10·3)

If we think of the problem in terms of path finding in the **Z** region of the collapsed N-state space, we can immediately see analogies with simple 'monkey problems'. These are problems suggested by McCarthy (1963), where a schedule of actions has to be found for a monkey that has to reach certain specified goals by moving in three-dimensional space, transferring objects from place to place, reaching objects, etc. It is clear that 'monkey problems' are simple prototypes of problems of reasoning about actions in the real world, such as assembling a physical object from parts, navigating a vehicle in a heavy traffic, etc. We can visualize our problem in the following way: a monkey is at the upper level of a two-level structure that has in its side an inclined stairway, and his goal is to reach a bunch of bananas that is at the lower level and at a certain distance from the stairway landing; suppose that the detailed geometry of the situation is as shown in the diagram of figure 10.2, where the scale of distances is in yards; suppose further that the monkey can always see the entire situation (the structure is essentially transparent): he can move over each level by using a 'horizontal' sequence of steps, he can move down the stairway by using a 'diagonal' sequence of steps, and he can safely jump vertical distances that do not exceed four yards; find a safe path that will bring the monkey to the bananas in the smallest number of steps. Clearly, the best solution trajectory for this monkey problem is isomorphic with the best solution that we have obtained for our original problem.

The solution of our illustrative problem (in any of the interpretations) would have been much more painful if the possible transitions were given as specifications of elementary steps. The availability of integrated, goal oriented, routines that specify macro-transitions is responsible for a substantial reduction in problem solving effort. A macro-transition is an expression of knowledge about the possibility of realizing certain sequences of transitions. It is a theorem about possible actions in the universe in which we are solving problems. Thus, the macro-transition (H_1,J_1) (see (10.1)) can be roughly interpreted in the 'monkey and bananas' context as asserting that it is possible for the monkey to go from *any* place on the upper level (except one corner point) to a place on the stairway which is four yards below the upper level. The proof of this assertion consists in exhibiting a sequence of realizable elementary steps that can be used by the monkey for going from any of the initial places at the upper level to the terminal place. Note that the elementary steps have themselves the status of macro-steps with respect to a lower level of possible actions. For example, in the M&C problem, we are using now a transfer across the river as an elementary step, and this transfer is realized by more elementary actions of loading the boat, moving it, and unloading it; in the 'monkey and bananas' interpretation, an elementary step may be realized in terms of certain sequences of muscle actions.

11. RELATIONSHIPS BETWEEN THE INITIAL SEARCH SPACE AND THE HIGHER LEVEL SEARCH SPACE

The high level space σ^* in which macro-schedules are constructed consists of a subset α of the set of $2(n+1)^2$ N-states, with the elements of α partially ordered under the attainability relation that is defined by the macro-transitions $\{(A)\}_5$ (given in (10.1)). The set α contains the following elements: the initial and terminal N-states that are specified in the problem formulation, and four N-states $(N,N,1)$, $(N-k+1,\ N-k+1,\ 1)$, $(0,0,0)$, and $(0,k,0)$ and the set of N-states $\{s|M_L=0,\ B_L=0,\ 0<C_L<k\}$. The initial or terminal N-states may coincide with some of the other elements; the set α has at most $5+k$ elements.

Let us examine the relationship between the new space σ^* and the space σ of $2(N+1)^2$ N-states. Consider the three sets $\{s\}_{top}=\{s|M_L=N\}$, $\{s\}_{diagonal}=\{s|M_L=C_L\}$ and $\{s\}_{bottom}=\{s|M_L=0\}$ in σ. They correspond to the top line, the diagonal line, and the bottom line respectively of the permissible **Z** region in σ. Each of these sets has one or more characteristic points that we call *entrance points* and the set of $\{s\}_{bottom}$ has a characteristic point that we call an *exit point*. The entrance point of $\{s\}_{top}$ is the initial N-state of the M&C problem, and the exit point of $\{s\}_{bottom}$ is the terminal N-state of the M&C problem; these are two elements of α. The entrance points of $\{s\}_{middle}$ are the N-states $(N,N,1)$ and $(N-k+1,\ N-k+1,\ 1)$; these are two elements of α (note that $(N,N,1)$ can be an entrance point of $\{s\}_{top}$ also). The entrance points of $\{s\}_{bottom}$ are $(0,0,0)$ $(0,k,0)$ and the points of the set $\{s|M_L=0,\ B_L=0,\ 0<C_L<k\}$; all of these are elements of α also. The macro-transitions (H_1) and (H_1,J_1) specify two possible ways of reaching an entrance point in $\{s\}_{middle}$ from an entrance point in $\{s\}_{top}$ The macro-transitions (D), (J_2), (D,J_2) specify three possible ways of reaching an entrance point in $\{s\}_{bottom}$ from an entrance point in $\{s\}_{middle}$. Finally, the macro-transition (H_2) specifies a way of reaching an exit point of $\{s\}_{bottom}$ from an entrance point in the same set.

We can think of the three sets $\{s\}$ as *easily traversable areas*, where a path for going from one point to another can be found with relative ease. However, the critical points of the problem occur at the points of transition, the '*narrows*', between easily traversable areas. These are represented by the intermediate entrance points. A substantial increase in problem solving power is obtained when such 'narrows' are identified, and when general ways of going from 'narrow' to 'narrow' are developed. Our macro-transitions provide precisely the capability of going from one 'narrow' into an easily traversable area, and then through that area to another 'narrow' that leads to the next easily traversable area or to the desired terminal exit.

The space σ^* is an *abstraction* of the space σ. Formally, a simplest solution to an M&C problem is attainable in σ^* if and only if it is attainable in σ. Furthermore, the minimal path linking two points in σ^* is identical with the minimal path between the same two points in σ. In σ^* attention is focused on a small number of well chosen critical points of σ. By looking for paths between points in σ^*, we solve the problem in at most three 'leaps', and then we can 'fill in' the details with the help of the definitions for the macro-transitions.

The main difficulty in finding an appropriate abstraction for the problem space lies in the discovery of the critical 'narrows' in that space, or more generally, of the topology of easily traversable areas and their connections in the problem space. After the 'narrows' are found, it is possible to build an abstract problem space that is based on them and that has ways of moving among them. It appears significant for the discovery of features in problem space–that lead to a formulation of an abstracted space–to have an appropriate representation of the space. Such is, we feel, the array representation that we have used for σ.

12. SUMMARY AND CONCLUDING COMMENTS

It is reasonable to expect that most 'real life' problems of reasoning about actions will not be formulated at the outset within a formal system. In many cases, the problem will have an initial verbal formulation. If such a problem is to be solved by a computer system, then the system must be able to accept a verbal formulation of the problem, and to convert this formulation into a form that is acceptable to a suitable problem solving subsystem. We have not considered in this paper the linguistic problem of *translating* from the natural language input into an 'internal' machine language that is acceptable to a problem solving program. This problem is receiving considerable attention at present (*see* Simmons, 1965). However, the question of choosing an 'appropriate' machine language, into which the verbal statement of the problem is to be translated, has received much less attention to date. In this paper, we are taking a first step towards understanding the nature of this question. Our notion of 'appropriateness' here is meant in the sense of suitability with respect to the *efficiency* of the problem solving process. In order to approach such a question of optimal choice of language, it is important to clarify the relationships between the language in which a problem is formulated for a problem solving system and the efficiency of the system. The systems of production P (introduced in section 2) provide a conceptual framework where such relationships can be studied. The 'internal' formulation of a problem amounts to specifying a system P, i.e. specifying the N-state language, the extended description language, the rules of action, and the two N-states that correspond to the initial and terminal situations between which the problem solving system is to find a solution trajectory. There exists considerable experience at present with computer realizations of problem solvers that work with formulations of problems in systems of production. GPS is an important prototype of such a problem solving system (see Newell, Shaw and Simon, 1960). To each system P there corresponds an N-state space over which the search for solution takes place. A good measure of the difficulty of the problem task is given by the size of the N-state space that must be searched to find a solution. Therefore, given a certain class of problems, we can evaluate the relative merits of languages for representing these problems in systems of productions by comparing the sizes of their associated N-state spaces that must be searched to obtain solutions.

In the specification of description languages for a system of productions where a given problem is to be formulated, the choice of basic elements (the universe U_0) and of basic predicates (properties and relations of the basic elements) is critical. This choice should provide enough expressive power for formulating the rules of action in a manner that reflects all the conditions of the problem. This is always possible if the elements and the predicates are chosen at a low enough, atomic, level; unfortunately, descriptions built of atomic elements have astronomical N-state spaces. Thus, we are confronted with the problem of finding the coarsest possible elements and predicates that can form descriptions that are fine enough for expressing the rules of action in the required detail. This is a difficult problem for people; at present, it is still more difficult for machines. In the M&C problem, we see that the initial formulation F_2 in a system of productions is much poorer than the formulation F_3 where instead of using individuals as elements, the sizes of certain sets of individuals (a much coarser notion) are considered to be the basic elements of the problem universe.

It appears desirable at present that an automatic translator whose task is to convert a verbal statement of a problem about actions to a machine formulation of the problem should have as its target language a language of descriptions that is atomic enough to accept quickly a great variety of problems about actions. The design of such a language seems possible and is now under study. The task of taking a possibly cumbersome system of productions P_1 from the output of such a translator and producing a better system P_2–in which the search for solution takes place–should then be delegated to the problem solving system. This is in accordance with our general thesis that it is an important function of the problem solver to find the most appropriate representation of his (its) problem. The separation of the initial translation process and the process of finding the most appropriate internal language for a problem appears to be methodologically desirable at present – given our state of knowledge about problem representations and conversions between them. It is conceivable, however, that the design of these two processes will be combined in the future. Undoubtedly, a unified approach to these two processes will strengthen both.

The rules of action of a system P play the role of the *laws of motion* that govern action sequences in the space of N-states. They are analogous to the differential equations that specify the possible time traces of a physical dynamic system. They are also analogous to the productions of combinatorial systems. Different types of problem conditions are reflected in different *forms* of rules of action. The non-cannibalism conditions of the M & C problem are easily expressible in the form of required derived descriptions for consequence of actions. As in the cases of differential equations and combinatorial systems, it is to be expected that there are classes of forms of rules of action to which there correspond problem spaces with certain special properties, characteristic patterns, etc. The identification and study of such classes would be an important contribution to the theory of problem solving processes. Even though such knowledge may not have direct implications for the design of problem solving systems that attempt to find a solution by intelligent search in a *given* problem space, it is most likely that it will be of great significance for the design of a system that would attempt to discover regularities in a problem space and that would subsequently use them for formulating new spaces where the process of searching for a solution becomes much easier.

An initial improvement in the formulation of the M & C problem came from the recognition that one of the conditions of the problem (non-cannibalism in the boat) is redundant. This permitted the formulation of new actions, as sequences of elementary actions, and it resulted in the effective elimination of many intermediate N-states. Hence, knowledge of the redundancy property permits a shrinkage of N-state space, i.e. an increase in problem solving efficiency. As shown in section 4, the redundancy of the boat condition can be established by deductive reasoning from the rules of action. Such reasoning can be carried out by machine theorem proving processes that are within the present state of the art. However, the process of *looking for* a redundant condition among the conditions of the problem is not a simple deductive process. It is a process of logical minimization. This also could be mechanized without much difficulty at present. The idea of eliminating redundant, irrelevant, conditions in a problem is an old and useful idea in the art of problem solving. It would pay then to have enough logical capabilities in a problem solving system in order to effectively attempt such eliminations.

In the M & C problem, an automatic conversion from the formulation F_2 to F_3 seems possible within the present state of the art. The conversion is based on the elimination of the redundant boat condition, the specification of compound transfer actions as sequences of the previous elementary actions (this is made possible by the previous elimination) and the formulation of new basic elements for the N-state language; this latter formulation can be guided by the form of the derived descriptions in the rules of action.

In section 6 we have shown that the formulation of the M & C problem in a production system is strongly equivalent to its formulation in a reduction system (which is a theorem proving system). A rule in the system of productions directly corresponds to a move (or a rule of inference) in the reduction system; the search trees are identical in the two systems. The reduction system has the advantage of showing clearly the logic of the attainability relations, as the search for solution evolves.

For each formulation of a problem in a system of productions it is always possible to specify an equivalent formulation in a reduction system. At worst, the search for solution in the reduction system will be identical with the search in the production system. In some cases, where the rules of action are *context free*, it is possible to specify stronger rules of inference in the reduction system, and to obtain as a consequence searches for solution that are faster than in a production system. A context free rule of action has the property that a given subconfiguration of an N-state can go to a specified subconfiguration of the next N-state regardless of the context of these subconfigurations in their respective N-states. In the M & C problem, the rules of action are strongly *context dependent*.

For example, no decision on the transfer of missionaries can be made independently of a decision on the transfer of cannibals or on the position of the boat. Thus, a reduction system cannot give an essential advantage in the M & C problem. An example where a reduction approach has considerable advantage for the solution of a problem that is formulated in a system of productions is the syntactic analysis of context free languages (*see* Amarel, 1965).

After the language of descriptions of a problem in a system of productions becomes reasonably efficient – as in the formulation F_3 in the M & C case – then the main improvements in problem solving power come from the discovery and exploitation of *useful properties* in the search space. An important property of this type is the *symmetry under time reversal* that we have found in the M & C problem. This property enables us to cut the depth of search for solution in N-state space by a factor of 2 – a significant reduction, hence a significant increase in problem solving efficiency. The symmetry property can be utilized by thinking in terms of a combined development of the search both from the initial N-state ahead in time, and from the terminal N-state back in time. However, only the development from one side is actually carried out. As soon as a search front reaches a point where there are linking possibilities between it and its image, then the search stops and a solution is found. In the present case, the formulation of the problem in a reduction system enables a clear development of the logic of search.

The symmetry property is strongly suggested by observing search graphs of the M & C problem (such as in figure 7.1) and also by examining the array representation of the N-state space. To establish the symmetry property (in section 8) we have used reasoning that is based on properties of the expressions for the rules of action. Again, such deductive reasoning is mechanizable at present. The mechanization of the more difficult task of *looking for* symmetries of certain type, given appropriate representations of solutions is also within sight. Given a newly discovered symmetry property, its utilization for problem solving requires reasoning *about* the problem solving process at a meta-level. This can be carried out with relative ease if the process is considered from the viewpoint of a reduction procedure and its logic interpretation.

In order to discover useful properties in the N-state space it is very important to have 'appropriate' representations of that space. In the M&C problem, the array representation (introduced in section 9) of N-state space has proved extremely fruitful. People have found the solution of M&C problems much easier when formulated as path finding in the array. Also, it is relatively easy for people to discover the properties that lead to the definition of macro-transitions. Is the 'appropriateness' of our array representation due solely to certain properties of the perceptual and reasoning processes of humans? Would this representation be as appropriate for (some) machine processes of pattern discovery? These remain open questions at present. In general, the problem of *choosing* a representation of N-state space, and of *discovering useful regularities* of solution trajectories in this representation, require much more study. Further exploration of these problems in the context of the 'dance floor' array representation of our M&C problem may provide interesting insights into them.

The definition of macro-transitions enables the formulation of the M&C problem in an extremely powerful system of productions (formulation F_6). The size of the N-state space is drastically reduced and a solution is obtained with practically no search, regardless of the size of the problem (sizes of populations to be transported and boat capacity). Macro-transitions act as well-chosen lemmas in a mathematical system; they summarize knowledge about the possibility of reaching certain critical intermediate points in the search space from some other points. The new N-state space that is based on macro-transitions is an *abstraction* of the previous N-state space. Only certain *critical points* of the lower level space appear in the abstracted space. We can reason in broad lines about the solution–and construct in the process a macro-schedule–by trying to establish a path, made of macro-transitions, that goes through some of these critical points. Once the macro-schedule is built, it is straightforward to obtain a detailed schedule by compiling the routines of action sequences that define the macro-transitions. The idea of finding a small set of points in the search space that are necessary and sufficient for the construction of the solution, is central in our last approach. In discussing the importance of such an approach, Simon (1966) brings the example of the simplex method in linear programming, where only the subspace made of the boundary points of the space of feasible points is searched for a solution.

The evolution of formulations of the M&C problem from its verbal statement to its last formulation in the abstracted subspace of the N-space is accompanied by a continuous and sizable increase in problem solving efficiency. This evolution demonstrates that the choice of appropriate representations is capable of having spectacular effects on problem solving efficiency. The realization of this evolution of formulations requires solutions to the following four types of problems:

(i) The choice of 'appropriate' basic elements and attributes for the N-state language.

(ii) The choice of 'appropriate' representations for rules of action and for the N-state space.

(iii) The discovery of useful properties of the problem that permit a reduction in size of the N-state space. Specifically, the discovery of a redundant condition in the problem, the discovery of symmetry in the problem space, and the discovery of critical points in the problem space that form a useful higher level subspace.

(iv) The utilization of new knowledge about problem properties in formulating better problem solving procedures.

Given solutions to (i) and (ii), it is conceivable that the approach to the solution of (iii) and (iv) is mechanizable–assuming good capabilities for deductive processing. There is very little knowledge at present about possible mechanizations of (i) and (ii). However, if experience in problems of type (iii) and (iv) is gained, then at least the notions of 'appropriateness' in (i) and (ii) will become clearer.

Acknowledgment

This work was greatly stimulated by discussions with A. Newell and H. A. Simon.

REFERENCES

Amarel, S. (1965), Problem solving procedures for efficient syntactic analysis, *ACM 20th National Conference.*

Amarel, S. (1967), An approach to heuristic problem solving and theorem proving in Propositional Calculus, *Computer Science and Systems.* Toronto: University of Toronto Press.

Bellman, R. & Dreyfus, S. (1962), *Applied Dynamic Programming.* Princeton: Princeton University Press.

Black, F. (1964), *A Deductive Question Answering System.* Unpublished Doctoral Dissertation, Harvard University.

Carnap, R. (1958), *Introduction to Symbolic Logic and its Application.* New York: Dover Publications.

McCarthy, J. (1963), Situations, a[illegible]nd causal law[illegible] [illegible]anford Artificial Intelligent Project, Memo No. 2.*

Newell, A. (1966), Some examples of problems with several problem spaces, *Seminar Notes*, CIT, Feb. 22.

Newell, A., Shaw, T. & Simon, H.A. (1960), Report on a General Problem-Solving program for computer, *Proceedings of the International Conference on Information Processing*, pp. 256–64. Paris: UNESCO.

Simmons, R.F. (1965), Answering English questions by computer: a survey, *Communications of the ACM*, **8**.

Simon, H.A. (1966), On reasoning about actions, CIT # 87, Carnegie Institute of Technology.

Simon, H.A. & Newell, A. (1961), Computer simulation of human thinking and problem solving. *Datamation*, June-July. 1961.