Tạp Chí Tin Học và Điều Khiển Học
Tập X(1994), Số 2, 21-28

A formula for creation and its applications to generate plots on computers

Doan Nguyen Hai

I. The creation formula

Among human intelligent activities, *creation* is always considered one of the superior forms. Creation, said briefly, is to produce somthing new in quality. To the question " Is Artificial Intelligence possible?", if we have some theory about machine creativity and several programs illustrating it, we will have more confidence and evidence to answer 'Yes' or even 'No'.

1) First we consider the extend of creation. To us, creation is both original and universal.

(a) In common view, creation is often thought to be original. It is somthing which occurs only once. Nobody has ever done it before, or the creating agent does not know whether so. Creation comes to us suddenly, in sense of we can not be sure of it. Thus creation contains much interesting secret.

(b) But creation is universal. Indeed, if we agree that all events in this universe occur uniquely, then every intelligent activity is creative. So it is a good idea to think of creativity as a fuzzy concept: all intelligent activities are creative, more or less.

Both the originality and the universality of creation prove the necessity to deal with creation in AI. Creation is universal, so AI machines should be able to do that everyday activity. Creation is original, so AI would be successful only if it can do that *high-quality* task.

2) The creation formula

$$\text{Creation} = \text{Knowledge} + \text{Logic} + \text{Incidentalness}.$$

Saying that creation has no formula is more acceptable than giving one to shape it. But we'd rather have a theory, a model, though not quite exact, than none.

We think that Creation is a *compound* of three substances which are Knowledge, Logic and Incidentalness. But let's analyse each of them.

(a) The first component, knowledge, will get easy agreement of most people. In general, to create, one need profound knowledge in the working domain. One of AI methodological sayings is 'Intelligence needs knowledge'. Profound knowledge helps the creative agent discover the problem nature and the characteristic relations in it to produce new things.

(b) But knowledge is not enough to create. We need a reasoning process to go from what we have had to what we have not, that is logic. Logic is the reasoning process, Good logic implies good results.

(c) If knowledge and logic require true work and talent, incidentalness seems to be the incarnation of unjustice. It exists in what we call good luck, what we call sudden. With the same talent and work, someone creates but the others don't. Or even the creator is of less talent and does less work.

All the substances demonstrate the originality and the universality of creation. Incidentalness surely implies the originality of creation but so do knowledge and good logic. Knowledge and logic control all intelligent activities but so does incidentalness because it is natural character of our universe and thus creation cannot be an exception.

II. The plot generation system

The Plot generation system is a software which is able to *generate plot of novel, play or film.* Its creative principle is our creation formula. Its purpose is to favor the argument that machine can create, at least when viewed at the result, and to illustrate the creation formula. The requiment for the plots generated is to be 'human', that means they must seem to be created by human, with logical, interesting, and some philosophical characteristics.

a.) The basis model

A plot consists of many events. At the begining, there some initial events - they can be entered by the user, or can be generated at random, or can be selected (by user or at random) from a list of initial event sets available in the system.

Ex: This initial set consists of 3 events:

A is rich. B is pretty. B wants to deceive A.

The generation of a plot is carried on in two directions:

(1) The developing direction: In this direction, events are generated consecutively as logical consequences of each other.

Ex: From the above set, these events could be generated:

A is rich. B is pretty. ⇒ *A loves B.*

A loves B. ⇒ *A and B marry.*

A and B marry. B wants to deceive A. ⇒ *B holds all A's property and then abandon A.*

B holds all A's property and then abadon A. ⇒ *A hates B.*

A hates B. ⇒ *A kills B.*

A kills B. ⇒ *B dies. A is arrested.*

(2) The expanding direction: In this direction, events are expanded into rich details. In this way, the plot becomes more interesting.

Ex: Expand some events:

A is rich. — *A is the boss of a big company*

B is pretty. → *B is the Miss of the town.*

A kills B. — *A poissons B.*

B dies. → *B is taken to the hospital but it is useless.*

The materials used to produce the events are knowledge taken from the knowledge bases. In general, each category (social, sentimental, science-fiction, ...) has its own knowledge bases because its category has particular charactiristics on the content of the events. One can even divide a category into subcategories each of which has its own knowledge bases.

The knowledge bases contain knowledge in two kinds of rules: *developing rules* and *expanding rules.*

Ex: Some developing rules:

A is rich. B is pretty. ⇒

A loves B. — *A loves B. B loves A.* — *A loves B. B does not love A.* — *A loves B. B loves only A's property.*

A loves B. B loves A. ⇒

A and B marry. A and B live happily. — *A and B marry. A and B live unhappily.* — *B dies. A is very unheppy.*

A loves B. ⇒

A and B marry. — *B does not love A. A is very unheppy.* — *B marries C. A hates B.*

Thus a developing rule has form of:

Cause set ⇒ *Result set 1* — *Result set 2* — ...

Ex: Some expanding rules:

A is rich → A is the boss of a big company. — A is a very rich man. — A's father is a millionaire.

A kills B. → A poisons B. — A shoots B. — A drowns B.

Thus an expanding rule has a form of:

An event → Set of detailled events 1. — Set of detailled events 2. — ...

The basic work of the plot generation goes in the developing direction. From the current event set (at the begining, it is the initial set), the system looks for an applicable developing rule, then apply it to generate new events by taking at random a result set on the righ-hand side of the rule.

In general, a rule is applicable to the current event set if there is a subset of this set which can be matched with the left-hand side of the rule. (Matching is tried between the names of characters in the current event set and the variables in the rule. A variable occurring in the righ-hand side but not in the left-hand side will corresponding to a new character). Moreover, there are other criteria:

(1) It is not acceptable a repetition of a pevious application. The reason is to avoid generating the same events again.

(2) A rule is applicable only if the characters matched with its variables are still alive.

(3) If there are more than one applicable rules the one with the longest left-hand side is preferred.

(4) Still, if ther are more than one applicable rules, the choice is at random.

The generation of events will finish when there are no more developing rules to apply or when the number of events generated has exceeded a predefined limit.

The last stage is to expand the events into detailled events using the expanding rules. An event is expanded by taking incidentally a set of detailled events on the righ-hand side of the matched rule. If there are no matched rules, the event is matched to itself. Finally, the detailled events are printed as the produced plot.

Principal characters

With the basic model, we will have unpredicted various plots, in each of which there could be some characters emerging as principal or there could be no principal characters at all. Now we propose a solution to set up principal characters before generation.

It can be thought simply that a character is *principal* when the proportion of events in which that character occurs to the total events (called the *occur proportion)* is greater than some predefined limit. In our experimental system, we have chosen

that limit = 60%. Similarly, we can establish the occur proportion of important subordinate characters from 40% to 60%.

The generation process now has some little additions:

(1) The principal character is established before the generation. And when his occur proportion in the current set is < 60%. the rule selected should generate new events such that this proportion gets ≥ 60%. Otherwise all goes normally. (Thus if the principal character has died byt his occur proportion is still ≥ 60%, more events could be generated as the tail of the plot. But, as this proportion decreases to 60% the process terminates.)

(2) Similar manipulation for important subordinate characters.

(3) With other characters the system sould control their occur proportions not exceeding the principal character's occur proportion.

b. The structural model

The basic model produces various plots not obeying any user's intention. For example, the system could give out a plot in which good people suffer much while evil-doers enjoy all pleasures. So now we propose the *structural model*, which is developed from the basic model and is able to generate plots following a structure.

For instance, we want to have plots with the philosophy '*Good defeats Evil*'. Such a philosophy can be expressed in the structures in the following structure rule:

Good defeats Evil ⇒

A is good and B is evil. B attacks A. A punished B successfully. —

A is good and B is evil. B attacks A. A punished B unsuccessfully. B attacks A. B punished A successfully. —

A is good and B is evil. B attacks A. A forgives B. B feels remorse.

This structure rule has tthree structures on the right-hand side. In general, our system can contain many structure rules for vatious philosophies (such as 'Evil defeats Good', 'Love overcomes all hatreds', 'Might is right', etc.) and each structure rule can have many structures. To produce a plot wich follows a philosophy, the system chooses incidentally a structure in the corresponding structure rule.

The events in a structure are called the *frame events*. For instance, the second structure in the above example consists of 5 frame events. We will use this structure to illustrate the generation process.

Suppose the initial set is:

A is rich. B is poor.

The generation process begins with the expansion of the first frame event. Then the developments and expansions are carried on as in the basic model. Next we continue all such things for the next frame event, then for the next gframe event,

..., till the end of the structure. If the structure has n frame events the generation process is a for-loop:

for i=1 to n do
begin
{ Stage i }
Expand the i'h frame event;
Develop and expand the events as in the basic model;
end;

The expansion of a frame event is due to the frame expanding rules.

Ex: Some frame expanding rules:

{ A is good and B is evel.
{ A is rich. B is poor. —

A is very kind and honest. B is ambitious and cruel. —
A often helps people. B is deceitful and ambitious.—
A often helps B but B is very ungrateful.

A is good and B is evil. ⇒
A is very kind and honest. B is wicked and ambitious.—
A always does good thing. B is selfish, ambitious and hypocritical.—

{ B attacks A.
{ A is rich. →

B robs A's property.— B deceives A to seize A's property.

B attacks A. ⇒
B harms A.— B insults A.— C is A's friend. B harms C.—
B wants to do evil things but A hinders B. B harms A.

And suppose we have the following developing and expanding rules:

B deceives A to seize A's property. →
B cheats A in playing cards and wins all A's property.—
B cooperates with A and deceives A to hold all A's capital.

B deceives A to seize A's property. ⇒
A becomes poor. B becomes rich.— A suicides. B becomes rich.— A becomes poor. B becomes rich. A hates B.

A becomes poor. B becomes rich. ⇒
C is A's lover. C abadons A to go with B.—
A works for B. B insults A.

Then we could have the first events of the plot as follow:

A is rich. B is poor. A often helps people. B is deceitful and ambitious. B cheats A in playing cards and wins all A's property. A becomes poor. B becomes rich. C is A's lover. C abandons A to go with B.

We have said that if the structure has n frame events the process will go through n stages. One important thing is the attachment betwwen the events in the previous stages and the succeeding satges. For instance, we'd rather have a frame structure rule like:

$$\left\{\begin{array}{l}\text{A punishes B unsuccessfully.}\\ \text{C is A's lover. C abandons A to go with B.}\end{array}\right. \Rightarrow$$

A fails to kill B but kills C instead.—
A just could kill B if C did not hinder A.

The attachment between stages sometimes requires a backtracking from a stage to its previous stages. For instance, if the stage '*B attacks A*' has led to the event '*A dies*' the process should backtrack to delete this event and to produce another one in order to be able to go on to the next stage ('*A punishes B unsuccessfully*'). However, there is also another way to avoid backtracking, by including the knowledge base rules like:

$$\left\{\begin{array}{l}A\ A\ punishes\ B\ unsuccessfully.\\ A\ dies.\end{array}\right. \Rightarrow$$

A is not really dead. A is saved by C. A punishes B unsuccessfully.—
A is incarnated into C. A punishes B unsuccessfully.

Other applications of the structural model

The structural model is not only applicable to generate plots with philosophical intentions but in general, it can be used to produce plots that should have a structure.

A good example is the detective plots. The basic model seems unsuitable for this kind which requires a good arrangement of events. But the structural model is promising:

Detective plots $\Rightarrow$

A is killed. B,C.D are suspect. B, C are proved innocent. D is proved the murderer.—

A is killed. B,C,D are suspect. B,C,D are proved the murderers.

Another application of the structural model is we can build structures having branching points. At the points, the plot may continue in one of many possibilities - the generation process may pause, list these possibilities and wait the user's choice.

Applications of the plot generation system

The plot generation system can be used

1) To prove the possibility that machine can do creative acts.

2) to generate plots og novel, pay or films for human writers.

3) to generate animated cartoons when combined with a drawing system.

The basic and structural models can be used to generate pictures and music.

The creation formula can be applied for other syustems doing creation and can be used as a theoretical model for creative activities.

III. For conclusion: the predestination motivation

Regarding human creation, there are many options about the motivation. Some think that it is the social motivation; to live and to develop, humanity must to creative acts. Some thinh that libido causes all. We do not consider what the motivation is here but we would like to present the idea that the motivation, whatever it be, is the one who has put the + and = signs into the creation equation.

Regarding machines, particularly the plot generation system, we call their motivation of creation a *predestination motivation*: machines do craetive acts because I have programmed them to do so. As a matter of fact, man man does not always why hi lives; is there any programmer who has programmed us to live, in his program, there is some looseness called incidentalness?

Abstract

In this paper we propose a formula for creation and present the design of a plot generation system as one application of this formula. Such a system has been implemented in our university in 1993.

Department of Computer Scinece, Polytechnical University of HoChiMinh City
268 Ly Thuong Kiet Str. HoChiMinh City.