HYPERSOLVER: A GRAPH-BASED TOOL FOR MODELING WITH SETS

Varol Akman and Mujdat Pakkan

BILKENT UNIVERSITY

Department of Computer Engineering and Information Science

Technical Report BU-CEIS-93-16

HYPERSOLVER: A Graph-Based Tool for Modeling with Sets

Varol Akman and Müjdat Pakkan
Department of Computer Engineering and Information Science
Bilkent University, Bilkent, 06533 Ankara, Turkey

1 Introduction

Set theory has long occupied a unique place in mathematics since it allows various other branches of mathematics to be formally defined within it [1]. The theory has ignited many debates on its nature and a number of different axiomatizations were developed to formalize its underlying 'philosophical' principles. Collecting entities into an abstraction for further thought (i.e., set construction) is an important process in mathematics, and this brings in assorted problems. The theory had many groundshaking crises (like the discovery of the Russell's Paradox [2]) throughout its history, which were nevertheless overcome by new axiomatizations.

The most popular of these is the Zermelo-Fraenkel axiomatization with 'Choice' (ZFC). ZFC is an elegant theory which inhabits a stable place among other axiomatizations as *the* mainstream set theory. It provides a 'hierarchical' framework. This hierarchy starts with only one abstract entity, the empty set ($\emptyset$), forms sets out of previously formed entities cumulatively, and is therefore called the *cumulative hierarchy*. The coherence of this hierarchy is secured by the Axiom of Foundation (FA) which forbids infinite descending sequences of sets under the membership relation $\in$, such as $\ldots \in a_2 \in a_1 \in a_0 \in a$ (thereby not allowing sets which can be constituents of themselves), and which has sometimes been regarded as a somewhat superficial limitation [2]. Sets which obey the FA are called *well-founded* sets.

The cumulative hierarchy has provided a precise framework for the formalization of many mathematical concepts [3]. However, it may be asked whether the hierarchy is limiting, in the sense that it might be omitting some sets one would like to have around. *Cyclic sets*, i.e., sets which can be members of themselves, are examples of such interesting sets which are excluded in ZFC. A set like $a = \{a\}$ is strictly banned in ZFC by the FA since a has no member disjoint from itself. Such sets have infinite descending membership sequences and are called *non-well-founded sets*. Non-well-founded sets have generally been neglected by the practicing mathematician since the classical well-founded universe was a satisfying domain for his practical concerns. However, non-well-founded sets are useful in modeling various phenomena in computer science, viz. concurrency, databases, artificial intelligence (AI), etc.

McCarthy stressed the feasibility of using set theory in AI and invited researchers to concentrate on the subject in a 1985 speech [4]. Circularity is an often exploited property in various fields of AI, e.g., commonsense reasoning. Rehearsing an example of Perlis [5], if non-profit organizations are considered as individuals, then the organization of all non-profit organizations is a set. It is conceivable that this umbrella organization (called NPO) might want to be a member of itself in order to benefit from having the status of a non-profit organization (e.g., tax exemption). But this implies that NPO must be non-well-founded, i.e., NPO $\in$ NPO.

This paper (also see [6]) investigates an alternative set theory (due to Peter Aczel [7]) which uses a graphical representation for sets and thereby allows the representation of non-well-founded sets. A program, called HYPERSOLVER, which can solve systems of equations defined in terms of sets in the universe of this new theory is presented.

2 Hyperset Theory

Hyperset Theory is an enrichment of the classical ZFC set theory. It is the collection of all the conventional axioms of ZFC modified to be consistent with the new universe involving atoms, except that the FA is now replaced by the AFA (to be explained in the sequel). The sets in this theory are collections of atoms (urelements) or other sets, whose *hereditary membership* relation can be depicted by graphs. These sets may be well-founded or non-well-founded, i.e., may have an infinite descending membership sequence, in which case they are also called *hypersets*.

Sets can be pictured by means of directed graphs in an unambiguous manner. For example, $a = \{b, \{c, d\}\}$ can be pictured by the graph in Figure 1. In this graph, each nonterminal node represents the set which contains the entities represented by the nodes below it. The edges of the graph stand for the hereditary membership relation such that an edge from a node n to a node m, denoted by $n \longrightarrow m$, means that m is a member of n. Since b, c, and d are assumed to contain no other entities as elements (i.e., they are urelements), there are no nodes below them.

In Aczel's terminology [7], a *pointed graph* is a directed graph with a specific node called its *point*. A pointed graph is said to be *accessible* if for every node n, there exists a path $n_0 \longrightarrow n_1 \longrightarrow \cdots \longrightarrow n$ from the point n_0 to n. If this path is always unique, then the pointed graph is a *tree* and the point is its *root*. Accessible pointed graphs (*apg*s) will be used to 'picture' sets.

A *decoration* D for a graph is an assignment of a set to each node of the graph in such a way that

$$D(n) = \begin{cases} \text{an atom or } \emptyset, & \text{if } n \text{ has no children,} \\ \{D(m) : n \longrightarrow m\}, & \text{otherwise.} \end{cases}$$

An apg G with point n is a *picture* of a if there exists a decoration $D(n) = a$, i.e., if a is the set that decorates the top node.

An apg is called *well-founded* if is has no infinite paths or cycles. *Mostowski's Collapsing Lemma* tells us that every well-founded graph has a unique decoration.

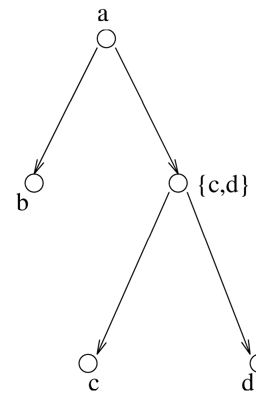

Figure 1: The graph representation of $a = \{b, \{c, d\}\}$

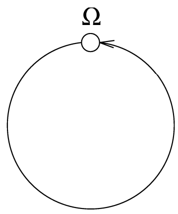

Figure 2: The picture of the non-well-founded set $\Omega = \{\Omega\}$

This leads to the corollary that every well-founded apg is a picture of a unique well-founded set. A non-well-founded apg can never picture a well-founded set because if a is the set which contains all the sets pictured by the nodes occurring in a cycle of the non-well-founded apg, then it can be seen that no member of a is disjoint from a itself, violating the FA.

Aczel's *Anti-Foundation Axiom (AFA)* states that every apg, well-founded or not, pictures a unique set, or stated in other words, every apg has a unique decoration [7]. AFA has two implications: existence and uniqueness. The former assures that every apg has a decoration (which leads to the existence of non-well-founded sets besides well-founded ones) and the latter asserts that no apg has more than one decoration. By throwing away the FA from the ZFC (and naming the resulting system ZFC$^-$) and adding the AFA we obtain the *Hyperset Theory* (a.k.a. ZFC$^-$/AFA).

One of the important advantages of the new theory is that by allowing arbitrary graphs, non-well-founded sets are included. For example, the non-well-founded set $\Omega = \{\Omega\}$ is pictured by the apg in Figure 2, and by the uniqueness property of the AFA, this is the only set pictured by that graph. Therefore, there is a unique set which is equal to its own singleton in the universe of hypersets.

The picture of a set can be *unfolded* into a tree picture of the same set. The tree whose nodes are the finite paths of the apg which start from the point of the apg,

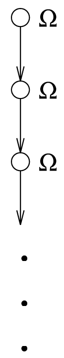

Figure 3: Unfolding Ω to obtain an infinite tree

whose edges are pairs of paths $\langle n_0 \longrightarrow \cdots \longrightarrow n\,,\, n_0 \longrightarrow \cdots \longrightarrow n \longrightarrow n' \rangle$, and whose root is the path n_0 of length one is called the *unfolding* of that apg. The unfolding of an apg always pictures any set pictured by that apg. Unfolding the apg in Figure 2 results in the infinite tree in Figure 3, analogous to unfolding $\Omega = \{\Omega\}$ to $\Omega = \{\{\{\cdots\}\}\}$.

The uniqueness property of AFA leads to an intriguing concept of extensionality for hypersets. The classical extensionality paradigm, that sets are equal if and only if they have the same members, works fine with well-founded sets. However, this is not of use in deciding the equality of say, $a = \{1, a\}$ and $b = \{1, b\}$ because it just asserts $a = b$ if and only if $a = b$ [8]. However, in the universe of hypersets, a is indeed equal to b since they are depicted by the same graph. To see this, consider a graph G and a decoration D assigning a to a node x of G, i.e., $D(x) = a$. Now consider the decoration D' exactly the same as D except that $D'(x) = b$. D' must also be a decoration for G. But by the uniqueness property of AFA, $D = D'$, so $D(x) = D'(x)$, and therefore $a = b$.

Aczel develops his own extensionality concept by introducing the notion of bisimulation. A *bisimulation* between two apg's, G_1 with point p_1 and G_2 with point p_2, is a relation $R \subseteq G_1 \times G_2$ satisfying the conditions

1. $p_1 R p_2$
2. if nRm then
   - for every edge $n \longrightarrow n'$ of G_1, there exists an edge $m \longrightarrow m'$ of G_2 such that $n'Rm'$
   - for every edge $m \longrightarrow m'$ of G_2, there exists an edge $n \longrightarrow n'$ of G_1 such that $n'Rm'$

Two apg's G_1 and G_2 are said to be *bisimilar*, denoted by $G_1 \sim G_2$, if a bisimulation exists between them; this means that they picture the same sets. It can be concluded that a set is completely determined by any graph which pictures it. Therefore, for

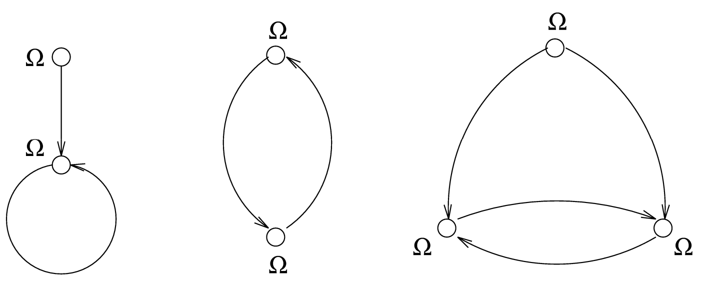

Figure 4: Other graphs depicting Ω

two sets to be different, there should be a genuine structural difference between them. For instance, the graphs in Figure 4 all depict the non-well-founded set Ω because their nodes can be decorated with Ω and there is no essential structural difference between them.

AFA has interesting applications. In [8], a modeling scheme for propositions (of natural language) is offered. In this scheme, the triple $\langle P, p, i\rangle$ denotes that the proposition p has the property P if $i = 1$, and it does not have it if $i = 0$. (In set theory, triples like $\langle x, y, z\rangle$ are defined as pairs of pairs, i.e., $\langle x, \langle y, z\rangle\rangle$, and $\langle y, z\rangle$ is defined as $\{\{y\}, \{y, z\}\}$.) If the proposition p is taken to be say, the statement

"This proposition is not expressible using eight words,"

then it can be modeled by the triple $\langle E, p, 0\rangle$ where E (an atom) is the property of being expressible (in English) using eight words. In Aczel's conception, p can be depicted as in Figure 5 where the longest arc shows that p refers to itself.

3 Solving Systems of Hyperset Equations

AFA has an important consequence which has useful applications allowing us to assert that some sets exist without having to picture them with graphs and which will be motivated by the following example [7].

An equation of the form $x = \langle 0, x\rangle$ in one variable x can be rewritten as $x = \{\{0\}, \{0, x\}\}$. This equation is equivalent to the following system of four equations in four unknowns:

$$\begin{aligned} x &= \{y, z\}, \\ y &= \{w\}, \\ z &= \{w, x\}, \\ w &= 0. \end{aligned}$$

By AFA, this system of equations has a unique solution pictured by the graph in Figure 6. Unfolding the original equation, one obtains $x = \langle 0, \langle 0, \langle 0, \ldots\rangle\rangle\rangle$. This result can be generalized. It can be shown that for any set a, the equation $x = \langle a, x\rangle$

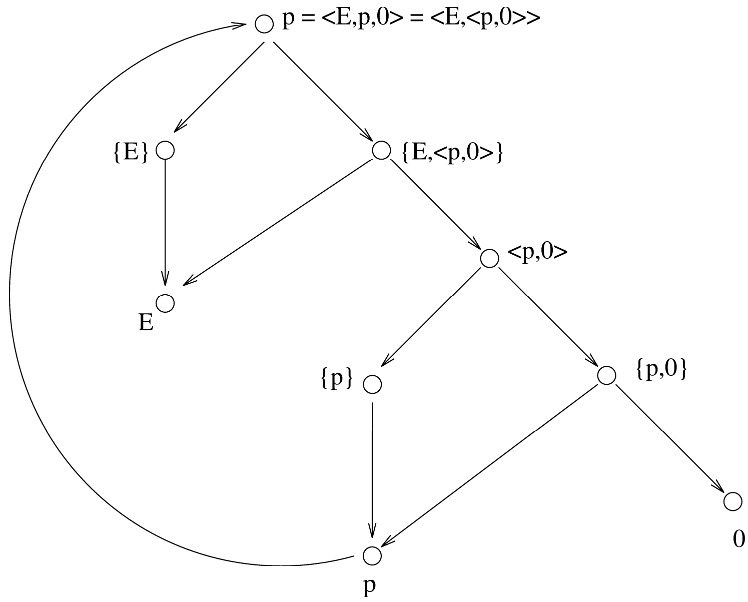

Figure 5: The Aczel picture of the proposition $p =$ "This proposition is not expressible in eight words"

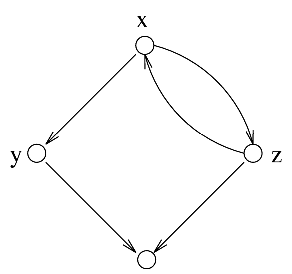

Figure 6: The solution of the system $x = \{y, z\}$, $y = \{w\}$, $z = \{w, x\}$, $w = 0$

has a unique solution $x = \langle a, \langle a, \langle a, \ldots\rangle\rangle\rangle$. More generally, if we consider an infinite system of equations

$$
\begin{aligned}
x_0 &= \langle a_0, x_1\rangle, \\
x_1 &= \langle a_1, x_2\rangle, \\
x_2 &= \langle a_2, x_3\rangle, \\
&\vdots
\end{aligned}
$$

then a unique solution

$$
\begin{aligned}
x_0 &= \langle a_0, \langle a_1, \langle a_2, \ldots\rangle\rangle\rangle, \\
x_1 &= \langle a_1, \langle a_2, \langle a_3, \ldots\rangle\rangle\rangle, \\
x_2 &= \langle a_2, \langle a_3, \langle a_4, \ldots\rangle\rangle\rangle, \\
&\vdots
\end{aligned}
$$

is seen to exist.

Motivated by such examples, a technique to assert that every system of equations has a unique solution has been developed by Aczel [7]. This technique is named the *Solution Lemma* by Barwise and Etchemendy [8] and is formulated below.

Let $\mathcal{V}_A$ be the universe of hypersets with atoms from a given set A and let $\mathcal{V}_{A'}$ be the universe of hypersets with atoms from another given set A' such that $A \subseteq A'$ and X is defined as $A' - A$. The elements of X can be considered as *indeterminates* ranging over the universe $\mathcal{V}_A$. The sets which can contain atoms from X in their construction are called *X-sets.* A *system of equations* is a set of equations

$$\{x = a_x \ : \ x \in X \ \wedge \ a_x \text{ is an } X\text{-set}\}$$

for each $x \in X$. For example, choosing $X = \{x, y, z\}$ and $A = \{\mathrm{C}, \mathrm{M}\}$ (thus $A' = \{x, y, z, \mathrm{C}, \mathrm{M}\}$), consider the system of equations

$$
\begin{aligned}
x &= \{\mathrm{C}, y\}, \\
y &= \{\mathrm{C}, z\}, \\
z &= \{\mathrm{M}, x\}.
\end{aligned}
$$

A *solution* to a system of equations is a family of pure sets b_x (sets which can have only sets but no atoms as elements), one for each $x \in X$, such that for each $x \in X$, $b_x = \pi a_x$. Here, π is a *substitution operation* (defined below) and πa is the pure set obtained from a by substituting b_x for each occurrence of an atom x in the construction of a.

The *Substitution Lemma* states that for each family of pure sets b_x $(x \in X)$, there exists a unique operation π which assigns a pure set πa to each *X-set* a, viz.

$$\pi a = \{\pi b \ : \ b \text{ is an } X\text{-set such that } b \in a\} \cup \{\pi x \ : \ x \in a \cap X\}.$$

The *Solution Lemma* can now be stated [7]. If a_x is an *X-set*, then the system of equations $x = a_x\,(x \in X)$ has a unique solution, i.e., a unique family of pure sets b_x such that for each $x \in X$, $b_x = \pi a_x$.

This lemma can be stated somewhat differently [9]. Letting X again be the set of indeterminates, g a function from X to 2^X, and h a function from X to A, there exists a unique function f for all $x \in X$ such that

$$f(x) = \{f(y) \,:\, y \in g(x)\} \cup h(x).$$

Obviously, $g(x)$ is the set of indeterminates and $h(x)$ is the set of atoms in each *X-set* a_x of an equation $x = a_x$. In the above example, $g(x) = \{y\}$, $g(y) = \{z\}$, $g(z) = \{x\}$, and $h(x) = \{\mathrm{C}\}$, $h(y) = \{\mathrm{C}\}$, $h(z) = \{\mathrm{M}\}$, and one can compute the solution

$$\begin{aligned} f(x) &= \{\mathrm{C}, \{\mathrm{C}, \{\mathrm{M}, x\}\}\}, \\ f(y) &= \{\mathrm{C}, \{\mathrm{M}, \{\mathrm{C}, y\}\}\}, \\ f(z) &= \{\mathrm{M}, \{\mathrm{C}, \{\mathrm{C}, z\}\}\}. \end{aligned}$$

The Solution Lemma is an elegant result, but not every system of equations has a solution. First of all, the equations have to be in the form suitable for the Solution Lemma. For example, a pair equations such as

$$\begin{aligned} x &= \{y, z\}, \\ y &= \{1, x\}, \end{aligned}$$

cannot be solved since it requires the solution to be stated in terms of the indeterminate z. (These are analogous to the Diophantine equations.) As another example, the equation

$$x = 2^x$$

cannot be solved because Cantor has proved (in ZFC^-) that there is no set which contains its own power set (no matter what axioms are added to ZFC^-).

This technique of solving equations in the universe of hypersets allows us to assert the existence of some sets (the solutions of the equations) without having to depict them with graphs. This feature can be of considerable help in modeling information which can be cast in the form of equations. An example concerning Situation Theory follows.

Situation Theory is a theory of meaning and information content developed by Barwise and Perry [10]. It tries to formalize a semantics for English in the way English speakers handle information. A *situation* is a limited portion of the reality. It can be taken as a whole interacting with other situations. An *infon* is an ordered list $\langle R, a, i\rangle$ where R is a relation, a is a proper sequence of arguments of R, and i is the polarity, taking 0 or 1 as its value. For a given R and a, only one of the two infons $\sigma = \langle R, a, 0\rangle$ or $\bar{\sigma} = \langle R, a, 1\rangle$ is a fact, namely the one which holds in some situation s. For example, the infon $\langle \mathit{sleeping}, \mathit{Tom}, \mathit{garden}, 1\rangle$ is a fact if and only if Tom is indeed sleeping in the garden. (As a notational convention, a polarity 1 is dropped.)

It is generally hypothesized that situations are sets of facts and therefore can be modeled by sets to make use of the existing set-theoretic techniques [11, 12]. Indeed, this was the approach Barwise and Perry chose in [10]. However, using Barwise's

Admissible Set Theory [3] as the principal mathematical tool in the beginning led to problems in the handling of circular situations and they had to turn to the Hyperset Theory [13]. To demonstrate this, an example concerning common knowledge will now be given, viz. the *Conway paradox* [14]. Two card players P_1 and P_2 are given some cards such that each gets an ace. Thus, both P_1 and P_2 know that the following is a fact:

$$\sigma \ : \ \text{Either } P_1 \text{ or } P_2 \text{ has an ace.}$$

When asked whether they knew if the other one had an ace or not, they both would answer 'no'. If they are told that at least one of them has an ace and asked the above question again, first they both would answer 'no'. But upon hearing P_1 answer 'no', P_2 would know that P_1 has an ace. Because, if P_1 does not know P_2 has an ace, having heard that at least one of them does, it can only be because P_1 has an ace. Obviously, P_1 would reason the same way, too. So, they would conclude that each has an ace. Therefore, being told that at least one of them has an ace must have added some information to the situation. How can being told a fact that each of them already knew increase their information? This is the Conway paradox. The solution relies on the fact that initially σ was known by each of them, but it was not *common knowledge*. Only after it became common knowledge, it gave more information.

Hence, common knowledge can be viewed as iterated knowledge of σ of the following form: P_1 knows σ, P_2 knows σ, P_1 knows P_2 knows σ, P_2 knows P_1 knows σ, and so on. This iteration can be represented by an infinite sequence of facts (where K is the relation 'knows' and s is the situation in which the above game takes place, hence $\sigma \in s$): $\langle K, P_1, s\rangle$, $\langle K, P_2, s\rangle$, $\langle K, P_1, \langle K, P_2, s\rangle\rangle$, $\langle K, P_2, \langle K, P_1, s\rangle\rangle$, ...

However, considering the system of equations

$$x = \{\langle K, P_1, y\rangle, \langle K, P_2, y\rangle\},$$
$$y = s \cup \{\langle K, P_1, y\rangle, \langle K, P_2, y\rangle\},$$

the Solution Lemma asserts the existence of the unique sets s' and $s \cup s'$ satisfying these equations, respectively, where

$$s' = \{\langle K, P_1, s \cup s'\rangle, \langle K, P_2, s \cup s'\rangle\}.$$

Then, the fact that s is common knowledge can more effectively be represented by s' which contains just two infons and is circular.

4 The Implementation

HYPERSOLVER is a stand-alone program which can solve equations in the universe of hypersets by making use of the Solution Lemma. It has built-in graphical capabilities for displaying the graphs depicting the equations input by the user and the solutions of these equations. HYPERSOLVER is implemented in Lucid Common Lisp. To communicate with the user and to display graphs, it makes use of the XView Window Toolkit built on the X Window System. The user interface of HYPERSOLVER, called the Command Interface, is shown in Figure 7.

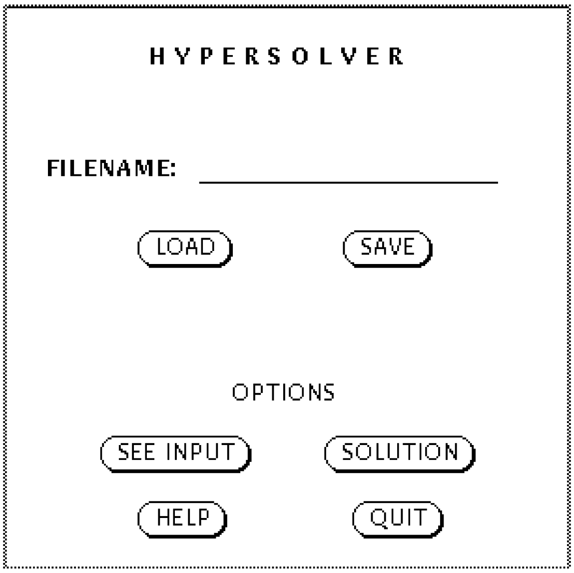

Figure 7: The Command Interface of HYPERSOLVER

4.1 Functionality

HYPERSOLVER solves a system of equations in the universe of hypersets. By a system of equations, the definition in Section 3 is meant:

$$\{x = a_x \ : \ x \in X \text{ and } a_x \text{ is an } X\text{-}set\,\}$$

for each $x \in X$, where X is a set of indeterminates, A is a set of atoms, and an X-*set* is a set which can contain elements from X. HYPERSOLVER does not solve systems which are not of this form. Therefore, taking $A = \{0, 1\}$ and $X = \{x, y\}$, a system like

$$x = \{0, 1, y\},$$
$$y = \{x\},$$

is a valid input for HYPERSOLVER, while the single equation

$$1 = \{x, y, 0\},$$

or the system

$$x = \{0, 1\},$$
$$x = \{x\},$$

are not since $1 \notin X$ and there should be a single equation for each $x \in X$. (HYPERSOLVER includes some filtering functions to detect invalid input.)

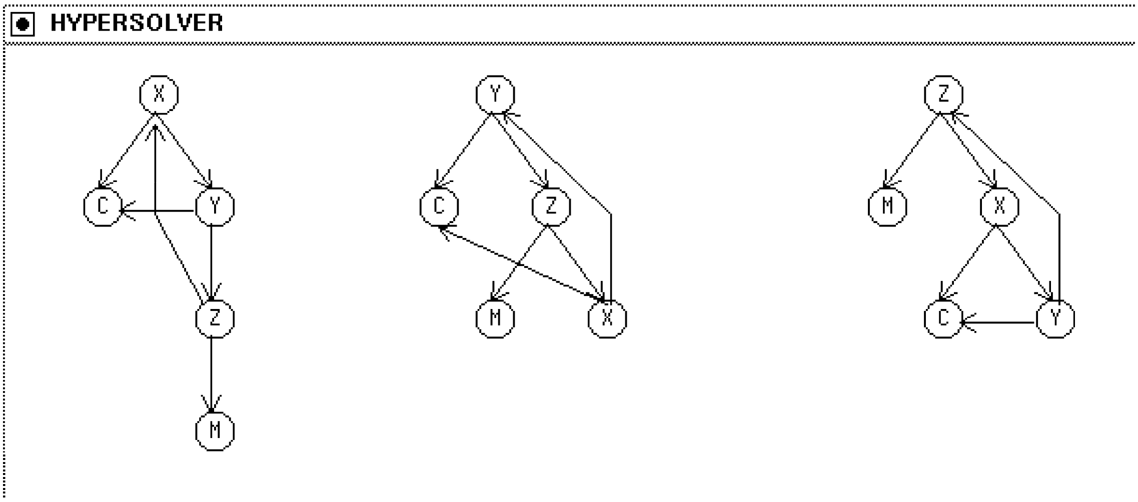

Figure 8: An example output of HYPERSOLVER

The notational conventions in HYPERSOLVER are as follows. Letters A through L are used to represent atoms of A, while letters M through Z represent indeterminates of X. The symbol @ will be used to represent the non-well-founded singleton Ω. (One-letter variable naming may seem quite limiting but it is simple to adopt the parser to handle variables with longer names.) Therefore, the graphs of solution given in Section 3 are depicted as in Figure 8.

HYPERSOLVER gets its input from a file which is to be specified by the user. The file must have one equation per line. For example, a file consisting of the following lines is a valid input file:

```
X={X,Y},
Y={A,B,Y,Z},
Z={X,Y,@}.
```

The input read from the file is sent to the parser of HYPERSOLVER. The parser is a character checking parser with a lookup table for the input characters. After converting the input into Lisp form, a transformation is applied to convert it to a list that can be processed by the equation solver. Finally, the input is checked to see whether it conforms the input requirements of HYPERSOLVER (e.g. if it contains one equation for each indeterminate, if each equation is of the form $x = a_x$, and so on).

The equation solving step of the HYPERSOLVER applies the Solution Lemma to the input system of equations. The alternative formulation mentioned in Section 3 is used for this purpose:

$$f(x) = \{f(y) \ : \ y \in g(x)\} \cup h(x),$$

for any set X of indeterminates where g is a function from X to 2^X and h is function from X to a set A of atoms. For the above example set of equations, $g(\mathrm{X}) = \{\mathrm{X}, \mathrm{Y}\}$,

$g(\mathrm{Y}) = \{\mathrm{Y},\mathrm{Z}\}$, $g(\mathrm{Z}) = \{\mathrm{X},\mathrm{Y}\}$ and $h(\mathrm{X}) = \emptyset$, $h(\mathrm{Y}) = \{\mathrm{A},\mathrm{B}\}$, $h(\mathrm{Z}) = \{@\} = @$. This representation scheme is suitable for recursive substitution. The algorithm of the equation solver performs this substitution by applying the Substitution Lemma on each equation of the input equation system. So, the solution for an indeterminate X can be found by finding the solutions of the indeterminates in $g(\mathrm{X})$ recursively. For each indeterminate, a decoration is found and the solutions are expressed in terms of these decorations. If the decoration for an indeterminate includes itself, then this denotes self-membership, and @ is used to signal that. For example, the decorations of the graphs for the above system of equations are (p, q, and r are the decorations for the indeterminates X, Y, and Z, respectively):

```
p={@,{A,B,@,{p,q,@}}},
q={A,B,@,{{@,q},q,@}},
r={{@,q},{A,B,@,r},@}.
```

To prevent duplicate substitutions which arise when an indeterminate occurs two or more times in an *X-set*, a list of already visited indeterminates is maintained. Nevertheless, because of the nature of recursion, duplication may occur in different levels of set nesting. Therefore, a kind of filtering is applied on the output of the solver to remove such duplicates.

The next step is the invocation of the graph display part of the HYPERSOLVER. This part takes the solution of a system of equations produced by the equation solver as input. As the general graph layout algorithm, a variant of the hierarchical layout algorithm proposed in [15] is exploited. The reason to use a hierarchical layout algorithm instead of a general-purpose algorithm is that most of the equations to be solved by the Solution Lemma will be hierarchical and that self-reference generally occurs for a single indeterminate. (Figure 5 is a good example of this.) A hierarchical algorithm leads to simplification in the display procedure and efficiency in run time.

The algorithm which has been adapted to the representation conventions and output requirements of HYPERSOLVER first forms the edge list of the solution system which consists of pairs of nodes. This list helps to get all children of each indeterminate. Then the nodes corresponding to these children are distributed to the levels taking care of the relationships between pairs of nodes. A more complicated part of the graph display unit is the one calculating the positions of the nodes on the screen. The hierarchical nature of the solution graphs is again exploited to make this calculation. The positions of the descendants of a node are calculated with respect to its own position, which in turn has been calculated with respect to its antecedents.

After the calculation of the positions, the actual graph drawing procedure is activated to display first the nodes and later the edges. This procedure pops up a large window (called the Graph Display Window, GDW) on which all graphical information is put. The output convention is such that the node labels which are the decorations of the sets represented by those nodes are written inside the node boundaries. While the edges which define hereditary membership are easily drawn, care has to be taken in case of a cycle. Cycles implying self-reference are not

Figure 9: The HYPERSOLVER graph of Ω

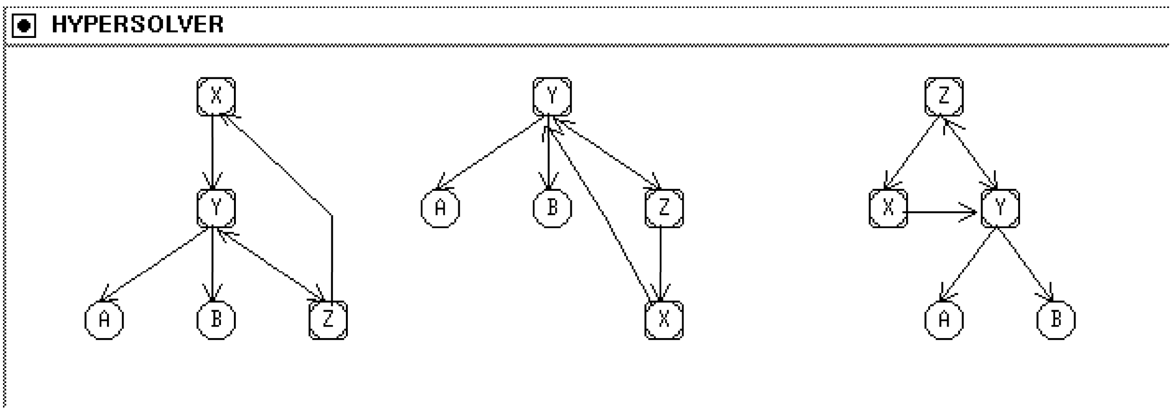

Figure 10: The graphs depicting the solution of the example in Section 4

displayed as circular edges, but are drawn in a different form. (Therefore, Ω is depicted as in Figure 9.) Cycles of one level are not much of a problem. If there exists a cycle between two nodes a and b, then the directed edge (b, a) can be drawn over the directed edge (a, b) to give a double arrow. However references to higher levels, especially to the root node representing the indeterminate are problematic since a path with minimum edge-crossing has to be found for aesthetic reasons. In such a case, paths walking around the graph are preferred (cf. Figure 12). Edge crossings may be unavoidable if no such path can be found. The solution graphs of the above example are depicted in Figure 10. The displaying of the graphs depicting the input sets proceeds exactly the same way as the displaying of the solution system. For example, the graphs of the input equations of the example system above can be seen in Figure 11.

4.2 Limitations and Future Work

HYPERSOLVER can solve any system which is in the form required by the Solution Lemma. This requires the equations to be in the form $x = a_x$ for each $x \in X$.

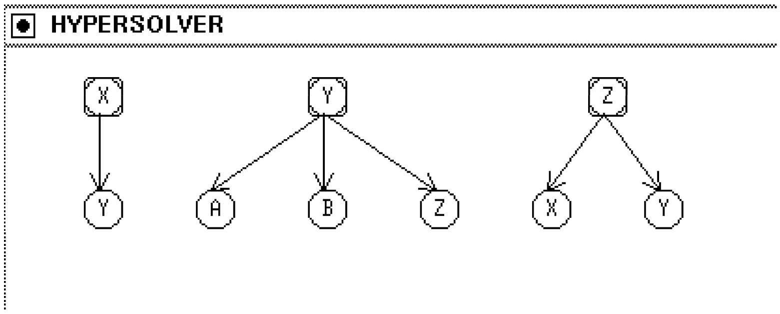

Figure 11: The graphs of the input equations of the example in Section 4

The systems which cannot be solved by HYPERSOLVER are those to which the Substitution Lemma cannot be applied. Such systems have been exemplified in Section 3.

HYPERSOLVER is generally weak in input/output operations. First of all it has limitations on the format of the input, such as one-letter variable naming, and one equation per line in the input file with no space between the characters of the input equations. These limitations arise because of the brittleness of the parser. A more powerful parser would let HYPERSOLVER be more flexible with input but the extra features would not add to the power of the program.

The graph display unit is another weak part of HYPERSOLVER. Graph drawing is a hard problem when considered for general graphs with any number of nodes [16]. Limiting the scope of the graph display problem as explained above reduces the difficulties considerably, but classical problems like minimizing the number of edge-crossings remain. HYPERSOLVER's graph display unit does not claim to know much about the graph layout problem. The algorithm does not work well for arbitrary graphs with no coherent node relationships. However, it works fine for the examples presented so far. Graph drawing problems are addressed in [17], [18], and [19] which propose generic graph browsers or editors.

Future work on HYPERSOLVER will concentrate on its applications to modeling of various phenomena in AI. This may include, for example, integrating HYPERSOLVER into a situation-theoretic framework [20] where the program may solve equations whose indeterminates can be unknown elements of situations, or unknown situations themselves. As a simple example, if a situation S is represented by the triple $\langle R, P, S' \rangle$, meaning object P is in relation R to another situation S', then S can be found in terms of S' by solving the equation $S = \langle R, P, S' \rangle$. Then, if S is a circular situation, P could also be in relation R to S itself, i.e., $S = \langle R, P, S \rangle$. This would, for example, correspond to an actual situation S in which a person P utters the statement "This is a very exciting situation." By "this situation," P is surely referring to the situation which his utterance describes. Such a circular situation S would be depicted as in Figure 12.

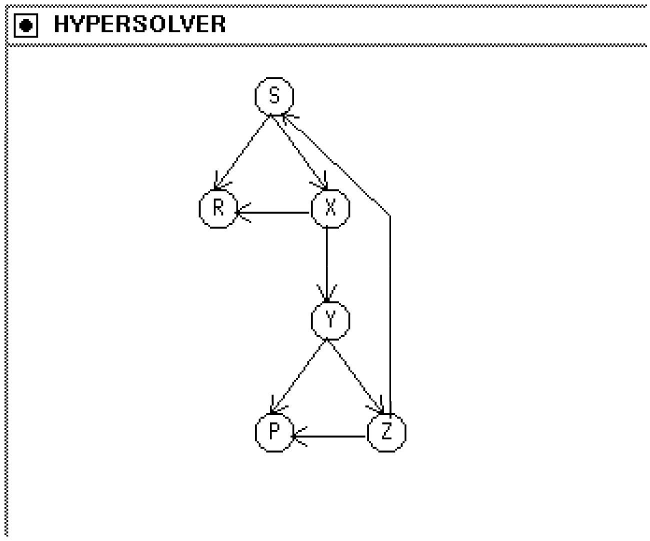

Figure 12: The graph of a circular situation $S = \langle R, P, S \rangle$

5 Conclusion

The Solution Lemma is a useful feature of the Hyperset Theory. Besides its mathematical importance and elegance, it provides an interesting way of modeling various circular phenomena.

The implementation presented in this paper, HYPERSOLVER, is a program which is based on the Solution Lemma and which can be a useful tool in areas of AI where information can be cast in the form of equations. Its simplicity, clarity, and well-defined user interface make it a practical instrument accessible for such purposes. When supported by a more general parser and a better graphical interface, it can be one of the emerging tools in mathematical logic, along the lines of, e.g., TARSKI'S WORLD [21].

HYPERSOLVER may be an important utility for *basic research* on the use of set theory in AI, too. Such research involving conceptual innovations is urgently needed in AI as pointed out by McCarthy in [22].

References

[1] P. Suppes. *Axiomatic Set Theory.* Dover, New York, 1972.

[2] A. A. Fraenkel, Y. Bar-Hillel, and A. Levy. *Foundations of Set Theory.* North-Holland, Amsterdam, 1973.

[3] J. Barwise. *Admissible Sets and Structures.* Springer-Verlag, Berlin, 1975.

[4] J. McCarthy. Acceptance Address, *International Joint Conference on Artificial Intelligence* (IJCAI) Award for Research Excellence, Los Angeles, 1985.

[5] D. Perlis. Commonsense Set Theory. In P. Maes and D. Nardi, editors, *Meta-Level Architectures and Reflection*, pages 87–98. Elsevier (North-Holland), Amsterdam, 1988.

[6] M. Pakkan and V. Akman. Issues in Commonsense Set Theory (Extended Abstract). In K. Oflazer, V. Akman, H. A. Güvenir, and U. Halıcı, editors, *Proceedings of the First Turkish Symposium on Artificial Intelligence and Neural Networks*, pages 47–52. Bilkent University, Ankara, 1992.

[7] P. Aczel. *Non-Well-Founded Sets.* Number 14 in CSLI Lecture Notes. Center for the Study of Language and Information, Stanford, CA, 1988.

[8] J. Barwise and J. Etchemendy. *The Liar: An Essay on Truth and Circularity.* Oxford University Press, New York, 1987.

[9] J. Barwise and L. Moss. Hypersets. *Mathematical Intelligencer*, 13(4):31–41, 1991.

[10] J. Barwise and J. Perry. *Situations and Attitudes.* MIT Press, Cambridge, MA, 1983.

[11] R. Parikh. Dumb-founded Sets. *Bulletin of EATCS*, 1989.

[12] V. Akman. Undaunted Sets. *ACM SIGACT News*, 23(1):47–48, 1992. Also in *Bulletin of EATCS*, 45:146–147, 1991.

[13] J. Barwise. Situations, Sets, and the Axiom of Foundation. In *The Situation in Logic*, number 17 in CSLI Lecture Notes, pages 177–200. Center for the Study of Language and Information, Stanford, CA, 1989.

[14] J. Barwise. On the Model Theory of Common Knowledge. In *The Situation in Logic*, number 17 in CSLI Lecture Notes, pages 201–220. Center for the Study of Language and Information, Stanford, CA, 1989.

[15] L. A. Rowe, M. Davis, E. Messinger, C. Meyer, C. Spirakis, and A. Tuan. A Browser for Directed Graphs. *Software—Practice and Experience*, 17(1):61–76, 1987.

[16] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications.* North Holland, Amsterdam, 1976.

[17] E. R. Gansner, S. C. North, and K. P. Vo. DAG—A Program that Draws Directed Graphs. *Software—Practice and Experience*, 18(11):1047–1062, 1988.

[18] T. Kamada and S. Kawai. An Algorithm for Drawing General Undirected Graphs. *Information Processing Letters*, 31:7–15, 1989.

[19] F. N. Paulisch and W. F. Tichy. EDGE: An Extensible Graph Editor. *Software—Practice and Experience*, 20(S1):63–88, 1990.

[20] E. Tın and V. Akman. BABY-SIT: Towards a Situation-Theoretic Computational Environment. *First International Conference on Mathematical Linguistics*, Barcelona, 1993.

[21] J. Barwise and J. Etchemendy. *The Language of First-Order Logic*. Number 23 in CSLI Lecture Notes. Center for the Study of Language and Information, Stanford, CA, 1990.

[22] J. McCarthy. Artificial Intelligence Needs More Emphasis on Basic Research: President's Quarterly Message. *AI Magazine*, 4(4):5, 1983.