Knowledge Acquisition Combining Analytical and Empirical Techniques

Ramón Sangüesa [2] Mario Martín [1] Ulises Cortés[1]

(1) Facultat d'Informàtica de Barcelona (UPC)
Departament de Llenguatges i Sistemes Informàtics (it)
Pau Gargallo, 5. Barcelona. 08028. Spain
e-mail: fibdls39@fib.upc.es ia@fib.upc.es

(2) Escola Universitaria d'Informatica de Lleida
Universitat de Barcelona
Estudi General de Lleida. Rambla d'Arago 307. 25006 Lleida. Spain
e-mail: fibdls02@fib.upc.es

ABSTRACT: In this paper we introduce a methodology for classification-oriented knowledge-base (KB) generation using LINNEO, a software for fuzzy classification and rule generation which resorts to analytical (Explanation Based Generalisation -EBG-) and empirical (Similarity Based Learning -SBL-) knowledge acquisition techniques. LINNEO builds a classification from a set of (frequently noisy) observations and a (possibly incomplete) domain theory supplied by the expert. The final result is a MILORD knowledge base. MILORD is a knowledge base generator which uses fuzzy rules for the management of uncertainty. It is believed that integrating both types (EBG, SBL) of learning techniques improves the whole process of knowledge acquistion. In our approach SBL is biased by a domain theory which helps in focusing the induction process. This is seen as a major advantage, overcoming the usual drawbacks of SBL techniques.

KEYWORDS: Automatic knowledge acquisition, machine learning, classification, expert systems, knowledge engineering, SBL, EBG, fuzzy logic.

1. INTRODUCTION.

One of the major problems in developing expert systems is knowledge acquisition. A variety of techniques have been devised in order to solve the "knowledge engineering bottleneck". A possible approach consists in using automated knowledge acquisition tools. These, in turn, make use of machine learning techniques, mainly Similarity Based Learning (SBL) and Explanation Based Generalisation

(EBG) [DEJO86], which stress two different paradigms in learning: inductive methods and deductive methods.

Inductive learning methods extract new hypothesis from observations but they need a great number of examples to be effective, they are generally blind to the relevance of certain attributes and the knowledge they finally infer may be erroneous. On the other hand, deductive methods ensure that new knowledge is logically sound. Moreover, they need much less examples. However, a consistent and complete domain theory is needed right from the beginning of the process which is not usually the case.

When one compares both approaches, one feels that, in a certain sense, they are complementary. It seems sensible to combine both knowledge acquisition techniques in order to overcome their disadvantages. Here we discuss LINNEO, a knowledge acquisition methodology in classification domains which integrates both techniques. On one hand, it uses - possibly noisy- observations to induce classes. On the other, it takes advantage of a domain theory, which represents the expert's current state of knowledge (so it may be incomplete), this theory constraints the possible outcomes of the classification process.

Figure 1 shows the general framework of the methodology. The human expert abstracts a collection of observations (sample) from the problem domain that he thinks of as relevant. He also selects a set of attributes (descriptors) that he assumes to be significant for the characterization of the domain objects. A sample is then modelled by a set of attribute/value vectors. At the same time, the expert can express what he already knows about the domain. This knowledge (that we will call Domain Theory) is represented as constraints and/or examples on classes already known to exist. Starting with this knowledge and informations LINNEO produces a classification. It uses induction over the observations taking into account the domain theory available which constrains the inductive process. The results of this classification are an intensive and extensive description of the generated classes and a fuzzy membership matrix that relates observations to generated classes. The expert evaluates the results and eventually modifies some attributes or some aspects of his domain theory. When a correct classification is obtained, rules are derived from the generated classes. Each rule includes in its antecedent the membership conditions and a fuzzy certainty factor for a particular class. Each antecedent is a conjunction of logical conditions on the values that descriptors must show to belong to the class whose identifier appears on the rule's consequent. These rules are analyzed and, through subsumption detection and synonymy analysis, they are organized in a hierachical way.

The results are made known to the user, who can accept and confirm them. He also has the chance to go back to the classification module, the definition of the Domain Theory or to the sample selection step and obtain another results. When all these actions are over, rules can be validated using new observations not previously in the sample. As soon as the expert accepts the resulting rule

set, it is transformed into MILORD [SIER89] rules and modules.

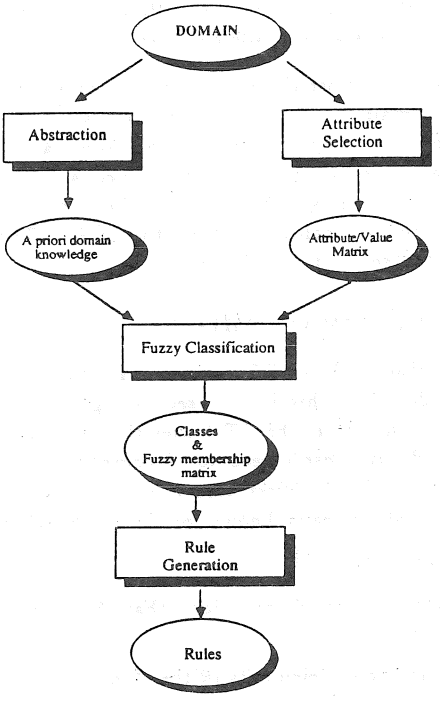

Figure 1. LINNEO: Overview of the different steps involved

2. ATTRIBUTE AND OBSERVATION SELECTION.

This is a very important step because it determines the quality of the results. Induction depends crucially on the quality of the given set of observations. In this step, the expert defines a set of observations that he/she thinks of as sufficient to model the domain and also defines a complete set of attributes relevant to the classification goal intended.

The attribute selection is essential because it determines the set of properties that define the set of objects. A poorly informative or barely discriminant attribute set returns an incorrect result. It should be noted that these properties will become the vocabulary used by the final expert system.

The expert is allowed to represent attributes by means of two pre-defined types:

- Quantitative: measurable properties. For example, in macroecnonomic classification, the national income, etc.
- Qualitative: properties that cannot be ordered. For example, the colour of an object, the marital status of someone, etc,.

It is also necessary to choose those observations considered important for the modelization of the domain. It is intended to cover all the possible prototipical cases. As it happened in attribute selection, an inadequate or incomplete set of observations may lead to a poor, when not erroneous, result. For each observation, a vector is defined whose length is just the number of attributes. This vector is the value vector for each property of the object. Then, we have a set of vectors representing the set of observations.

3. EXPRESSING THE DOMAIN THEORY.

In this step a domain theory (DT) expressing what the expert can explain about the domain is defined. This DT is used as a group of constraints guiding the inductive process. Therefore, the DT biases the set of possibles classes. This DT, just acts as a guide; it doesn't need to be complete.

The expert is allowed to express this theory in terms of constraints and examples on classes he/she already knows for sure that exist:

- Class name: an identifier.
- Constraints on classes: set of conditions that elements must fulfill in order to belong to the class.
- Examples: a list of typical elements of the class.

Conditions appearing in constraints need not correspond to the attributes used in defining observations. However, there should exist a mapping between the former and the latter. This mapping is expressed by means of a dictionary which can roughly be understood as a set of rewriting rules. On the other hand, examples have to be a subset of the previously defined observations.

The importance of defininig these constraints and examples is related to its usefulness in guiding the classification process.

4. CLASSIFICATION STEP.

Once the expert has selected the descriptors and the observation sample and has defined the corresponding constraints (in case he/she knows them), classification starts. This process induces a tentative conceptual structure for the domain. In general, any inductive classification process will group objects into classes using some criterion on similarity (SBL technique). In our approach, constraints on *a priori* classes are added to this similarity criterion.

We also want to set similarity within the framework of fuzzy sets theory. With these ideas in mind we have decided to use the classical concept of distance as a fuzzy similarity value. If we take d* as the normalized distance (d) then s*, is defined as:

$$s^* = 1 - d^*$$

that is, the fuzzy similarity value. Note that fuzzy similarity and distance are mutually inverse: the greater the distance the lower the similarity.

As we have seen in section 3., objects are represented as numeric/symbolic vectors of length p, p being the number of descriptors in the observation matrix. Position i of vector j shows the symbolical or numerical value of descriptor i of object j. The distance that will be used in determining the similarity between two objects is the general Hamming distance so that the distance between object O_i and object O_j is:

$$d(O_i, O_j) = \sum_{k=1}^{n} (dif(O_{ik}, O_{jk}))$$

Where:

- dif (O_{ik}, O_{jk}): if k is a qualitative property, its value is 1 if O_{ik} and O_{jk} are different modalities and 0 otherwise.
- dif (O_{ik}, O_{jk}): if k is quantitative is the absolute value of their difference

Two aspects are taken into account when representing a class: its extensional description and its intensional description. The first one is given by the enumeration of all elements contained in the class. The second one is a vector with p properties containing also the class centre and the standard deviation. Let's see an example:

```
(WEALTHY-COUNTRIES)
EXAMPLES: (W-GERMANY)
CONSTRAINTS:
    (AND (> EXECUTIVE-SAFETY 0.6) (INVESTMENT-TREND EXCELLENT))
ELEMENTS:
    (SOUTH-AFRICA W-GERMANY AUSTRALIA CANADA DANMARK JAPAN
NORWAY NETHERLAND SINGAPUR SWITZERLAND)
CENTRE:
    ((EXECUTIVE-SAFETY 0.8019)
    (FINANCIAL-SAFETY 0.7784)
    (POLITICAL-SAFETY 0.7858)
```

(PRESENT-INVESTMENT (LOW 0.2) (GOOD 0.2) (STABLE 0.6))
(INVESTMENT-TREND (LOW 0.0) (GOOD 0.0) (STABLE 0.0) (EXCELLENT 1.0))))
STANDARD DEVIATION: (0.2 0.1 0.12 NIL NIL)
MAXIMUM-MINIMUM: ((0.953 0.679) (0.896 0.712) (0.822 0.757) NIL NIL)

This is a example of an *a priori* class. Its existence, name, two constraints and one example are known and have been explicited by the expert. The rest of the fields are completed during the classification process.

The center of the class is obtained by calculating the mean value for each quantitative property of every object. For qualitative properties, the center includes each one of their modalities with its corresponding occurrence frequency. Note hat the center of a class can be considered as a prototype of the objects contained in the class. Therefore, it is possible to compare any object against a class. The distance between an object and a class can be taken as the inverse of the degree of fuzzy membership of the object to the class.

If a value for the distance is set as the limit for an object to belong to a class (limit that we will call radius), then the intended classification rule (expressing the similarity criterium with the DT) is found:

IF *the distance between a given object and the center of a class is less than the class radius* **AND** *it fulfills the class constraints*
THEN *it can belong to the class*
ELSE *it cannot belong to it.*

In this context, the radius r can be interpreted as the degree of selectivity of the classification. If the radius is large, there will be few classes with many observations in each one. If it is small, there will be many classes containing few observations. A tradeoff has to be found between large clusters of observations (large radius) and a large number of classes avoiding low-similarity elements to be clustered together (small radius). Frequently, the expert will have to create several classifications with different radiuses until he/she being able to choose one of them.

Taking this criterion into account we have developed the algorithm that we use for classification. Its outline is as follows:

Initialization:

- *Load the classes a priori defined by the expert.*
- *Verify that the examples of the classes fullfill the decision rule.*
- *Initialize the centre of the a priori classes with the examples. The initial center of a class is just a vector with the mean of the properties of the examples.*

Body: For each one of the objects in the system the best class among the current ones is selected:

- *Select those classes whose constraints are not violated by the object.*
- *The best class for an object is taken to be the one in the previous set of classes with minimum distance to the object. Two things can happen at this moment:*

  a) Distance to the best class is less than the current classification radius. In such case, the object is included in the class and the center of the class is recalculated. While recalculating the centre some objects may go away from the centre and locate themselves farther than the radius. If this happens, these objects are eliminated from the class to be be reclassified later but this time without modifying the centers.

  b) Distance to the best class is greater than the current classification radius or there exists not a best class. In such case a new class is created. The vector of the object currently under consideration becomes the initial centre of the class.

As it can be seen, induction is guided by the DT expressed as constraints. In our apporach, inductive (SBL) and deductive (EBG) are the extreme points of a single continous axis. At one extreme, when no DT is available, the algorithm works as a pure SBL method. At the oppposite one, when a complete and consistent DT is at hand, the algorithm is merely a deductive technique (EBG) with no induction at all. At the middle of this axis we have the situation when induction is biased by a DT (possibly not complete).

When the expert agrees with no classification, he/she will probably have to modify the set of observations, the set of atributtes or perhaps the domain theory previously defined. This is a usual situation that corresponds to a cyclical interaction between the user and the system in order to define effectively the domain. Verification of classification quality can be done with a collection of functions that calculate distances between class centers (in order to spot class overlappings), between centers and objects (to see if some element located under the radius of two classes is badly assigned), etc.

Similarly, besides expert opinion, several criteria can be used to decide when a given classification can be tagged as "good":

- Every class contains a similar number of observations.
- The number of useful descriptors is small.
- Significative distance exists between class centers (greater than their radiuses)
- Few overlappings (observations falling under two classes)
- etc.

The result, once it has been confirmed by the expert, is a list of classes and a fuzzy membership matrix. This two items of information are then transferred

to the next module wich will turn them into classification rules.

Once a partition over the observations is obtained, the same algorithm can be applied on the class centers, obtaining then a superclassification, a hierarchical classification tree.

The classification module here discussed has several advantages:

- It takes advantages of empirical and analitical techniques
- It obtains in a fast way a first aproximation to the domain structure.
- It works incrementally: addition of new data does not imply reclassification of previous observations.
- It deals simultaneously with numerical and categorical attributes.
- It works with prototypes (and an intensional description of classes).

5. THE RULE GENERATION MODULE

This module obtains a set of classification rules which have to mimic as approximately as possible the classifier's behaviour. The generated rules have the following structure:

$$(R_i, A, C, \alpha)$$

where,

R_i: rule identifier

A (antecedent): A logical conjunction of conditions that have to be met by observations to be included in a given class. For example,

$$(color(red)) \wedge (weight(high))$$

C: (consequent): The identifier of one of the classes previously obtained.

α : the degree in which the observations belong to the class. It is the certainty value associated with the rule's conclusion.

In order to obtain the rules, the description of classes and the fuzzy membership matrix created by the classfier are used. These two information items are the final result of the classification module. The first one allows us to elicitate which conditions have to be met by any observation to be included in each class. The second one allows calculation of the certainty level at which one can say that the observation belongs to a class. Recall that the center of a class is a vector C whose dimension is equal to the number p of properties defined by the expert. Each property has an associatetd value. In the case of a quantitative property,

the corresponding value of the centre is the mean value of this property calculated over all objects in the class. For qualitative properties the center contains each one of it modalities and the corresponding frequency within the class. This frequenciy gives an idea about how significant is the modality within the class. The "center" indicates which values must have the properties for a given object to be considered as part of the class.

The rule generator turns the values of the centre into logical expressions that correspond to the antecedent of a rule. In the case of qualitative properties this transformation is straightforward A predicate is created. Its name is the same as the property. This predicate is applied to the value of the modality appearing in the centre. If, for example, we have a series of observations about esponges and one of the characteristics used is "cortex" with modalities "present" and "absent", they will be transformed into the predicates `cortex(present)` and `cortex(absent)`. Of course, this conversion will not be always made. Previously one has to decide if this characteristic is relevant or not. The heuristic we use in detecting the relevance of descriptors is very simple but effective. In principle, the center always being in normalized form, we consider values above some value as "high" and under some other value as "low". Both values have been previously set by the expert. Then, values above the first value and under the second one are taken to be significant.

It is important to realize that values very near to 0 may suggest that what is really significant is the absence of the characteristic they refer to. Therefore, they will be transformed into negated predicates. For example, in the country classification example, if we consider the "trend" property whose modality "excellent" has a normalized value of 0.0 the corresponding transformation is: `(trend(not(excellent))`. Finally it has to be noted that negative characterization of values near zero is not always necessary. This is done when no other significant descriptor has been found or when anbiguities arise between two or more rules sharing the same significant descriptors and the negative characterization is the only way to discriminate between them.

Uncertainty factors assessment:

Each rule has an associated degree of certainty, calculated as the degree of membership to the class of observations that it is characterized by the rule's antecedent. This information is drawn from the fuzzy membership matrix previously obtained by the classification process. If the number of objects is n and the number of classes m, the fuzzy membership matrix is an $n \times m$ matrix of numerical values ranging form 0 to 1. Theses values show the degreee of membership for

every object to every class.

	Class 1	Class 2	Class 3
Elem. 1	1.0	0.2	0.1
Elem. 2	0.6	0.3	1.0
Elem. 3	0.4	1.0	0.8
Elem. 4	1.0	0.8	0.45
Elem. 5	1.0	0.8	0.7

$\alpha = 0.55$

Figure 3. Fuzzy membership matrix.

For an α value of 0.55 the elements of the different classes are shown by a shadowed square. Of course, the interesting α values are those that induce a change in the structure of classes. In general, the relation between class aggregation and the α values is given by a decreasing nonmonotonic function. Starting with an α value that ensures a convenient certainty (0.55, for example) and incresasing it is easy to see that the structure of classes changes, reducing the number of their elements. Using this property we get rules of this kind:

$$(r_1, A_1, C_1, 0.55)$$

$$(r_2, A_2, C_1, 0.65)$$

$$(r_3, A_3, C_1, 0.85)$$

$$(r_4, A_4, C_1, 0.95)$$

$$(r_5, A_5, C_1, 1.0)$$

being the α values precisely those values that changed the structure of the class. All these rules point to the same class. Their antecedents, however, are differnet because their components have also changed with α and, accordingly, the calculated centre for each α has been different. This has originated different antecedents, that is, different membership conditions for each class. In general the lower the α value the less strict the conditions will be. In other words, rule specificity increases with the α value. Having several rules with increasing certainty for the same class gives us a very interesting information. Less certainty

implies greater generality. This is the base for organizing rules in a hierarchical way. The fundaments of the process are described in [MART90a].

Analysis and diagnostic

As we explained in the introduction, the rule generation module offers several tools for analysis and diagnostic. The expert can use them to refine the rules in the rule generation module or he/she may choose to go back to the classification module. The analysis currently made by the rule classification module are:

- Synonymy between groups of descriptors. The distances bewtween descriptors are found. If there is zero distance between a pair of descriptors, they are considered to be synonymous. The user is asked to confirm this and a name is requested for the new substituing descriptor.
- Rewriting of equivalent descriptors. Sometimes, it is possible to check if the distances between several descriptors and another different one is very small. After user's confirmation the second one substitutes the first group.

To sum up, the rule generation process involves the following steps:

Certainty factors calculation

Starting from the class representation generated by the classifying module and from the fuzzy membership matrix:

$$(clase_1(c_1, ..., c_m, q_1, ..., q_n))$$

$$(clase_k(c_1, ..., c_m, q_1, ..., q_n))$$

$c_1, ..., c_m$ being the values of quantitative properties $q_1, ..., q_n$ being those of qualitative properties.

Following iterating over the α values, several centres are obtained for each class and an "intermediate" rule representation is generated.

$$((c_1, ..., c_m, q_1, ..., q_n)\alpha_{11}\, class_1)$$

$$((c_1, ..., c_m, q_1, ..., q_n)\alpha_{12}\, class_1)$$

$$((c_1, ..., c_m, q_1, ..., q_n)\alpha_{j1}\, class_j)$$

$$((c_1, ..., c_m, q_1, ..., q_n)\alpha_{kq} class_k)$$

Significant property selection

Rules in intermediate representation are scanned marking all descriptors whose value is above or under the levels set by the user. A new set of rules is obtained, each rule now having a marked descriptors list added.

$$((c_1, ..., c_m, q_1, ..., q_n)\alpha_{kq} class_k(c_3\, c_5\, q_2\, q_5))$$

Antecedent ambiguity elimination

All pairs of rules that share the same marked descriptors are found. A search for discriminating descriptors is started. First, descriptors with value near to zero appearing in one rule but not in the other are looked for. If they are found, they are inserted in the marked descriptors list. If the ambiguity cannot be solved, the user is warned and may review the current classification starting againg classification and rule generation with new parameters.

Antecedent simplification

Once ambiguities are solved, antecedents are substituted by the marked descriptors lists. Similarly an interactive process starts trying to cluster mutually synonymous descriptors (through distance analysis). If they are found, and after user's confirmation, a rewriting rule of the following form is created:

$$(c_1, c_2, ..., q_j) \Longrightarrow (c_i, c_k, q_r)$$

where $\Longrightarrow$ is the rewriting symbol so that any new rule having $(c_1, c_2, ..., q_j)$ in its antecedent will be changed having as new antecedent (c_i, c_k, q_r).

Transformation into predicate representation

After obtaining an adequate set of nonambiguous rules, the transformation into predicate representation can start. The final result is a collection of rules of the form:

$$(pred_1(value_1) \wedge pred_2(value_2) \wedge ... \wedge pred_3(value_3), \alpha, class_i)$$

These rules are then translated into MILORD formalism becomig a knowledge base that can be directly used by this expert system shell.

6. CONCLUSIONS

LINNEO, a software for rule generation in classification domains has been discussed. Its main contribution is the integration of two different learning techniques in knowledge acquisition. Integration tries to overcome the shortcomings of SBL and EBG techniques when acting on their own. With LINNEO the expert doesn't need to have a complete and fully consistent theory of the domain to start gathering knowledge. Nevertheless, the expert is able to express any previous knowledge he/she may have about the structure of the domain. This can be done by means of constraints and examples. In the former state the constraints are necessary conditions for objects to belong to a given (and known to exist) class. The latter facilitate the characterization of prototypical description of classes. In our view SBL and EBG can be seen as complementary poles in the same single axe. LINNEO II allows to use pure extremes or mixed techniques according to the explicited knowledge by the expert.

LINNEO has been successfully applied to sponge classification [DOMI90], mental illness classification [ROJO91] and economical classification [MART90b].

7. REFERENCES

[AGUI86] Aguilar, J., Piera, N. *"Les connectifs mixtes: de nouveaux operateurs d'associations des variables dans la classification automatique avec apprentisage"* In: **Data Analysis and Informatics**, edited by E. Diday. Elsevier Science Pub. (1986). pp. 253-265.

[CORT86] Cortés, U., López de Mántaras, R., Agustí, J., Plaza, E., *"Fuzzy knowledge engineering techniques in scientific document classification."* ACM Sigart. International Symposium on Methodologies for Intelligent Systems pp.94-112. Knoxville, Tennesse.

[DEJO86] DeJong, G.F, Mooney, R.L. *"Explanation Based Learning: an alternative view"* Machine Learning, 1 (1986).

[DOMI90] Domingo, M. *"Aplicació de tècniques de I.A. (LINNEO) a la classificació sistemática: O. Hadromerida (Demospongiæ.Porifera)"* Master Thesis. Ecology Dept. University of Barcelona (1990).

[LOPE89] López de Mántaras, R., Plaza, E. *"Model-based knowledge acquisition for Heuristic Classification Systems"* SIGART Newsletter. Special issue on Knowledge Acquisition, 108 (Abril 1989).

[MART90a] Martín, M. *CCA: Conceptual Connectivity Analysis. Une application pour l'analyse de descriptions conceptuelles floues.* Report Laboratoire d'Automatique et Analyse de Systemes. Toulouse, 1990 (in press).

[MART90b] Martín, M. Sangësa, R. *LINNEO: Herramineta para la adqusición de conocimientos y generación de reglas en dominios de clasificación y diagnosis.* IBERAMIA-90. Morelia, MEXIC, 1990.

[MICH85] Michalski, R, Steep, R. E. *"Learning from observation: Conceptual Clustering"* **Machine Learning an A.I. perspective**, Tioga, Palo Alto, CA. (1983).

[PIER87] Piera, N. **Connectius de lògiques no estandàrd com a operadors d'agregació en classificació multivariable i reconeixement de formes.** Ph.D. Facultat d'Informàtica de Barcelona. Universitat Politècnica de Catalunya. Barcelona (1987).

[PLAZ87] Plaza, E. **Sistema d'ajut a l'adquisició i estructuració de coneixements** Ph.D. Facultat d'Informàtica de Barcelona. Universitat Politècnica de Catalunya. Barcelona (1987).

[QUIN83] Quinlan, J.R. *"Learning Efficient Classification Procedures"* **Machine Learning an A.I. perspective**, Tioga, Palo Alto, CA. (1983).

[ROJO91] Rojo, E. *"Aplicació del software LINNEO a la classificació d'enfermetats mentals"* Ph.D. (forthcoming) University of Barcelona (1990).

[SIER89] Sierra, C. **MILORD. Arquitectura multi-nivell per a sistemes experts en classificació.** Ph.D. Facultat d'Informatica de Barcelona. Universitat Politècnica de Catalunya. Barcelona (1989).

[ZADE65] Zadeh, L. *"Fuzzy sets"*. Information and Control, 8, pp. 338-353, (1965).