**UNIT 3 Logical Modeling**

3.1 Decision Table

A decision table is a scheduled rule logic entry, in table format, that consists of conditions, represented in the row and column headings, and actions, represented as the intersection points of the conditional cases in the table. Decision tables are best suited for business rules that have multiple conditions. Adding another condition is done by simply adding another row or column.

Like the if/then rule set, the decision table is driven by the interaction of conditions and actions. The main difference is that in a decision table, the action is decided by more than one condition, and more than one action can be associated with each set of conditions. If the conditions are met, then the corresponding action or actions are performed.

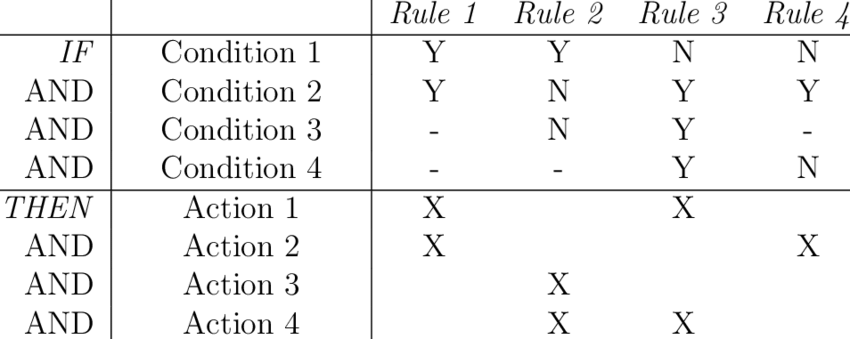


Fig. Decision table

3.2 Decision Tree

Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

 Fig.A decision tree for the concept PlayTennis.

**Construction of Decision Tree:** A tree can be *“learned”* by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called*recursive partitioning*. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of a decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high-dimensional data. In general decision tree classifier has good accuracy.

3.3 Structure English

It is similar to our structured programming. It uses logical construction and imperative sentences to carry out instructions for action. Decisions are made through IF, THEN, ELSE and SO statements. The structured English for the above ex: the publishers discount policy is

We can actually make structured English more compact by using terms defined in the data dictionary. For ex: the process ORDER may have the data element ORDER-SIZE that defines four values.

**MINIMUM**: 5 or fewer copies per book title.   
**SMALL**: 6 to 19 copies   
**MEDIUM**: 20 to 49 copies   
**LARGE:** 50 or more   
  
Using these values, the Structured English will be   
  
COMPUTE-DISCOUNT   
IF order is from bookstore   
    And-IF ORDER-SIZE is SMALL   
           THEN: Discount is 25%   
    ELSE (ORDER-SIZE is MINIMUM)   
           SO: no discount is allowed   
    ELSE (order is from libraries or individual customer)   
           So-if ORDER-SIZE is LARGE   
                 Discount is 15%   
           ELSE IF ORDER-SIZE is MEDIUM   
                 Discount is 10%   
           ELSE IF ORDER-SIZE is SMALL   
                 Discount is 5%   
           ELSE (ORDER-SIZE is MINIMUM)   
                 SO: no discount is allowed

3.4 Data Dictionary

A data dictionary is a file or a set of files that includes a database's metadata. The data dictionary hold records about other objects in the database, such as data ownership, data relationships to other objects, and other data. The data dictionary is an essential component of any relational database. Ironically, because of its importance, it is invisible to most database users. Typically, only database administrators interact with the data dictionary.

The data dictionary, in general, includes information about the following:

* Name of the data item
* Aliases
* Description/purpose
* Related data items
* Range of values
* Data structure definition/Forms

The **name of the data item** is self-explanatory.

**Aliases** include other names by which this data item is called DEO for Data Entry Operator and DR for Deputy Registrar.

**Description/purpose** is a textual description of what the data item is used for or why it exists.

**Related data items** capture relationships between data items e.g., total\_marks must always equal to internal\_marks plus external\_marks.

**Range of values** records all possible values, e.g. total marks must be positive and between 0 to 100.

**Data structure Forms:** Data flows capture the name of processes that generate or receive the data items. If the data item is primitive, then data structure form captures the physical structures of the data item. If the data is itself a data aggregate, then data structure form capture the composition of the data items in terms of other data items.

The mathematical operators used within the data dictionary are defined in the table:

|  |  |
| --- | --- |
| **Notations** | **Meaning** |
| x=a+b | x includes of data elements a and b. |
| x=[a/b] | x includes of either data elements a or b. |
| x=a x | includes of optimal data elements a. |
| x=y[a] | x includes of y or more occurrences of data element a |
| x=[a]z | x includes of z or fewer occurrences of data element a |
| x=y[a]z | x includes of some occurrences of data element a which are between y and z. |