Suppose we wanted to build a data mining system

Data mining, requires:

background knowledge;

concept hierarchies; and

representation of the expected results.

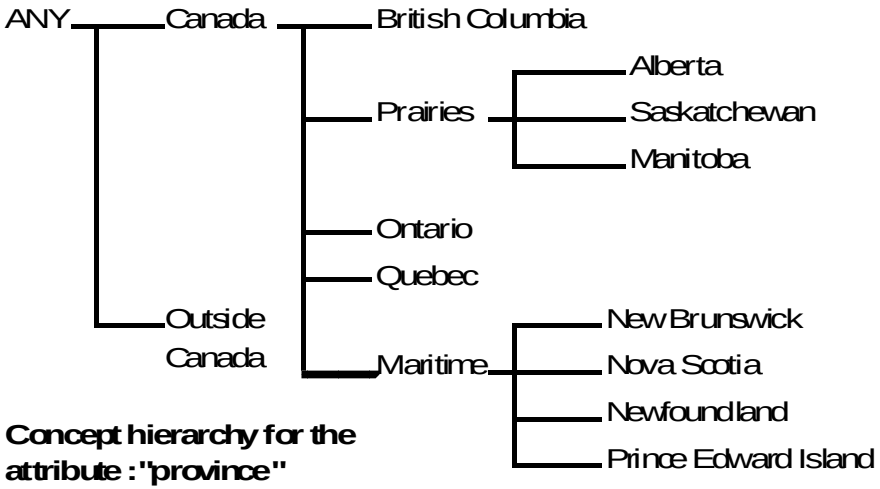
What can data mining do?

1. Classification
   discrete outcomes (bird, cat, or fish)
2. Estimation
   continuously valued outcomes (height, income, or weight)
3. Prediction
   estimated future value (predicting the number of children in the next year, or which telephone subscribers will order a three-way calling)
4. Affinity Grouping (market basket analysis)
   grouping which things go together (pizza and soft drink, coffee and coffee maid, or cat food and kitty litter)
5. Clustering
   segmenting a heterogeneous population into a number of more homogeneous subgroups (a cluster of symptoms might indicate different diseases)
6. Description
   describing what is going on (women support Democrats in greater numbers than do men)

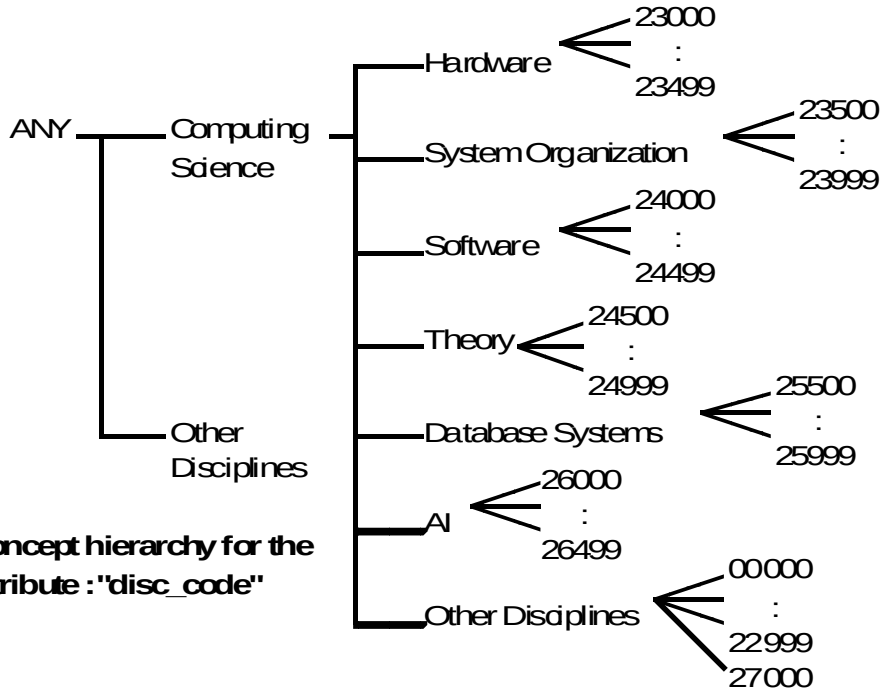
Background Knowledge

Name	Sex	Age	Birth_Place	Department	Position	Salary
Anderson	female	26	Burnaby	cmpt	secretary	26000
Bach	male	38	Ottawa	electr_engr	lab_manager	41000
Barton	male	30	Toronto	chem	junior_lecturer	28000
Benson	female	45	Vancouver	cmpt	full_prof	63000
.....	.....	...	.....	.....	.....	.....
Winton	male	38	Seattle	civil_engr	assoc_prof	55400
Young	male	55	Bonn	german	full prof	68000

Concept Hierarchies



Concept hierarchy for the attribute : "province"



Concept hierarchy for the attribute : "disc_code"

Representation of the learning results

Each tuple in a relation is considered a logical formula in conjunctive normal form and a data relation is considered as a disjunction of these. The data for learning and the rules discovered can be represented in either relational form (a table) or first-order predicate calculus, e.g.,

$$\forall x\ \text{graduate}(x) \rightarrow$$
$$\{\text{Birth\_Place}(x) \in \text{Canada} \ \&\ \text{GPA}(x) \in \text{excellent}\}\ [75\%]\ \vee$$
$$\{\text{Major}(x) \in \text{science} \ \wedge\ \text{Birth\_Place}(x) \in \text{foreign} \ \wedge$$
$$\text{GPA}(x) \in \text{good}\}\ [25\%].$$

Early Data Mining – DBLEARN

QUERY: learn characteristic rule for "CS_Op_Grants"
from Award A, Organization O, grant_type G
where O.org_code=A.org_code and
G.Grant_order="Operating Grants"
and A.grant_code=G.grant_code and
A.disc_code="Computer"
in relevance to amount, province, prop(votes)*,
prop(amount)

prop() is a built-in function which returns the number of original tuples covered by a generalized tuple in the final result and the proportion of the specified attribute respectively.

using table threshold 18

Result of query: early DBLEARN provided tabular output

Amount	Geography Area	# of Grants	Prop. of amount
0-20Ks	B.C.	7.4%	4.7%
0-20Ks	Prairies	8.3%	5.4%
0-20Ks	Quebec	13.8%	8.7%
0-20Ks	Ontario	24.5%	15.7%
0-20Ks	Maritime		
20Ks-40Ks	B.C.	5.3%	7.0%
20Ks-40Ks	Prairies	5.3%	6.6%
20Ks-40Ks	Quebec	5.1%	7.0%
20Ks-40ks	Ontario	12.9%	16.0%
20Ks-40ks	Maritime	1.0%	1.3%
40Ks-60Ks	B.C.	1.2%	3.1%
40Ks-60Ks	Prairies	0.2%	0.4%
40Ks-60Ks	Quebec	1.0%	2.5%
40Ks-60Ks	Ontario	5.1%	11.5%
60Ks-	B.C.	0.2%	0.6%
60Ks-	Prairies	0.4%	1.6%
60Ks-	Quebec	0.2%	0.6%
60Ks-	Ontario	1.2%	4.5%
Total:	$10,196,692	100%	100%

Several points are worth noting at this time:

The general framework presented for data mining has been attribute-oriented generalization.

Attribute-oriented generalization takes advantage of the organization of relational database systems.

The concept tree ascending technique follows from version spaces, typical of learning from examples paradigms.

Version space employ tuple-oriented generalization; DBLEARN's method uses concept hierarchies of each attribute as a factored version space and performs generalization on individual attributes, significantly increasing processing efficiency.

If there are p nodes in each concept tree and k concept trees (attributes) in the relation, the total size of the factored version space should be p^k - attribute-oriented generalization thus has a much smaller search space than tuple-oriented generalization.

Basic Attribute-Oriented Generalization Algorithm for Relational Databases

Input: (i) A relational database, (ii) a concept hierarchy table, (iii) the learning task, and optionally, (iv) the preferred concept hierarchies, and (v) the preferred form to express learning results.

Output: A characteristic rule learned from the database.

Method: Basic attribute-oriented induction consists of the following four steps:

Step 1. Collection of the task-relevant data.

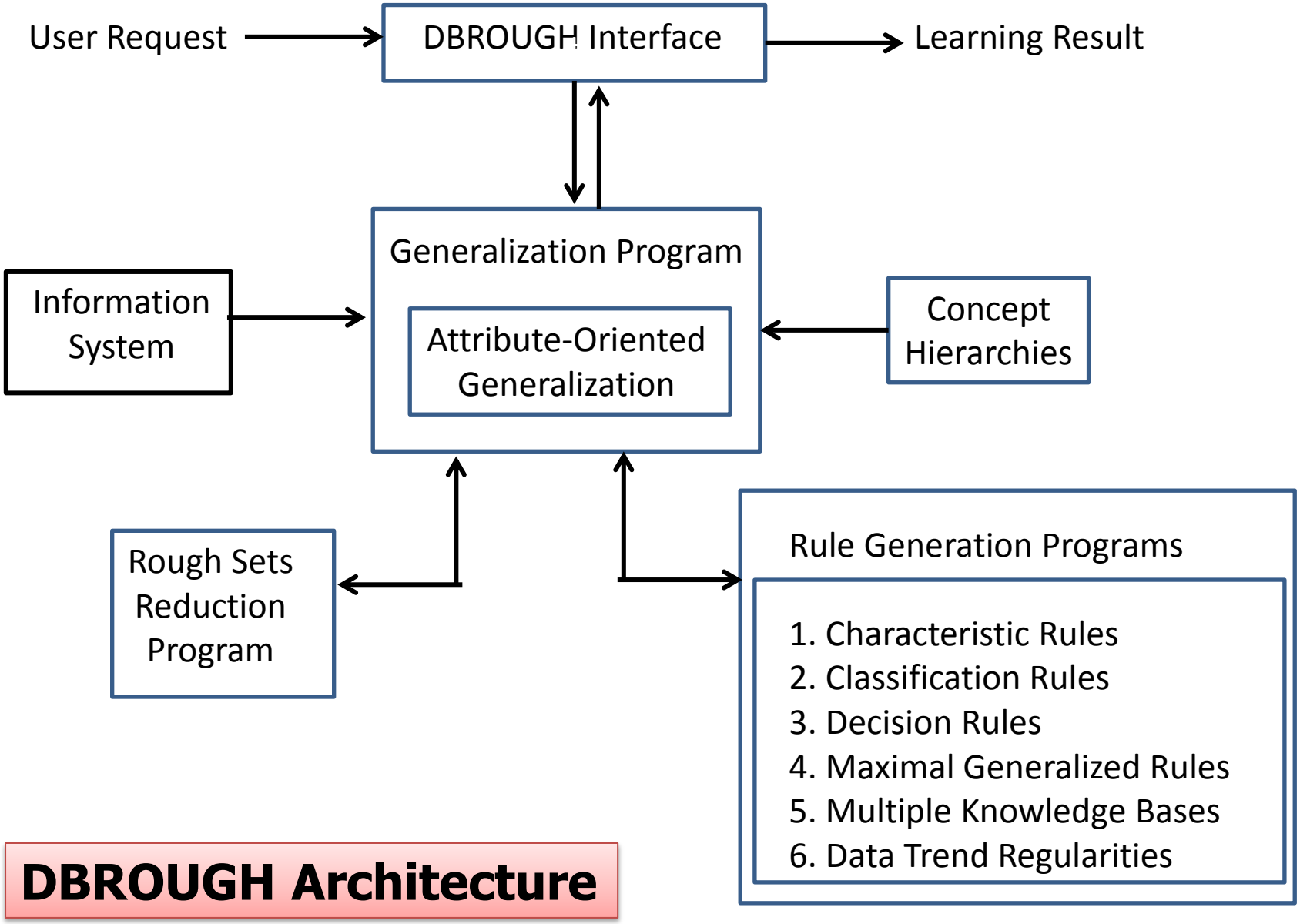
Step 2. Basic attribute-oriented generalization.

Step 3. Simplification of the generalized relation, and

Step 4. Transformation of the final relation into a logical rule.

Notice that the basic attribute-oriented generalization (Step 2) is performed as follows.

begin for each attribute Ai (1<i<n, # of attributes) in the generalized relation **do**
  while number_of_distinct_values_in_Ai > generalization_threshold **do**
  begin **if** no higher level concept in the concept hierarchy table for Ai
    then remove Ai
    else substitute for the values of Ai's by its corresponding minimal generalized concept;
    merge identical tuples
  end
  while #_tuples_in_generalized_relation > generalization_threshold **do**
    selectively generalize some attributes and merge identical tuples
end. {Basic attribute-oriented generalization}



DBROUGH Architecture

DBROUGH Actions

Extraction of the Prime Relation from a set of data R

Feature Table T_A extraction for an attribute A from the generalized relation R'

Attribute oriented generalization for discovering characteristic and equality rules w/a concept hierarchy.

Extract a generalized information system from a relation (EGIS).

Compute a reduct (GENRED)

Compute a set of maximally generalized rules (GENRULES)

The Prime Relation

Using rough set theory we can analyze the attributes globally and identify the most relevant attributes to the learning task.

The learning phase consists of two phases, data generalization and data reduction. In data generalization, the method generalizes the data by performing attribute-oriented concept tree ascension to obtain a Prime Relation. The generalized prime relation contains only a small number of tuples and it is feasible to apply rough set techniques to eliminate the unimportant or irrelevant attributes and choose the best minimal attribute set.

In the data reduction phase the method finds a minimal set of interesting attributes that have all the essential information of the generalized relation, thus the minimal set of attributes can be used instead of the whole attribute set of the generalized relation.

A Prime Relation Rp for a set of data R stored in a relational table is an intermediate relation generalized from relation R by removing non-desirable attributes and generalizing each attribute to a desirable level.

Algorithm: Extraction of a prime relation

Input: 1. a set of task relevant data R (obtained by a sql query), a relation of arity n with a set of attributes Ai ($1 \leq i \leq n$)

2. a set of concept hierarchies Hi

3. a set of desirability thresholds Ti for each attribute Ai

Output: the prime relation Rp

Method: 1. Rt=R (temporary variable)

2. for each attribute Ai or Ri do

if Ai is non-desirable then remove Ai

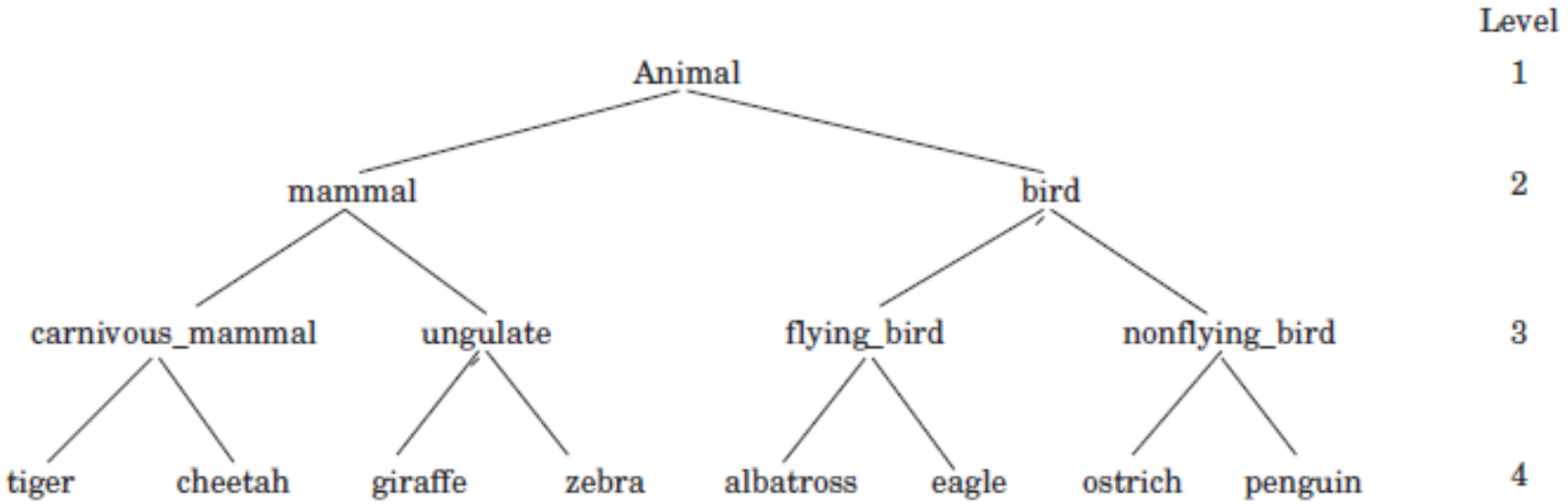
if Ai is not desirable but generalizable then generalize Ai to the next level

{generalization: collect the distinct values in the relation and compute the lowest desirable level L in which the number of distinct values will be no more than Ti by synchronously ascending the concept hierarchy from these values. Generalize the attribute to this level L by substituting for each value Ai's with its corresponding concept Hi at level L.}

3. Rp=Rt. (after identical tuples are merged)

Label	Animal	Hair	Teeth	Eye	Feather	Feet	Eat	Milk	Fly	Swim
T11	tiger	Y	pointed	forward	N	claw	meat	Y	N	Y
HA1	cheetah	Y	pointed	forward	N	claw	meat	Y	N	Y
FT3	giraffe	Y	blunted	side	N	hoof	grass	Y	N	Y
HJ8	zebra	Y	blunted	side	N	hoof	grass	Y	N	Y
O9H	ostrich	N	N	side	Y	claw	grain	N	N	N
KJ2	penguin	N	N	side	Y	web	fish	N	N	N
OL2	albatross	N	N	side	Y	claw	grain	N	Y	N
LP1	eagle	N	N	forward	Y	claw	meat	N	Y	N
TT1	viper	N	pointed	forward	N	N	meat	N	N	N

An animal world



Conceptual hierarchy of the animal world

Animal	Hair	Teeth	Eye	Feather	Feet	Eat	Milk	Fly	Swim	Vote
cmammal	Y	pointed	forward	N	claw	meat	Y	N	Y	2
ungulate	Y	blunted	side	N	hoof	grass	Y	N	Y	2
nonflyb	N	N	side	Y	claw	grain	N	N	N	1
nonflyb	N	N	side	Y	web	fish	N	N	N	1
flying	N	N	side	Y	claw	grain	N	Y	N	1
flying	N	N	forward	Y	claw	meat	N	Y	N	1
viper	N	pointed	forward	N	N	meat	N	N	N	1

The prime relation table

Start with a generalized relation

Animal	Hair	Teeth	Eye	Feather	Feet	Eat	Milk	Fly	Swim	Vote
mammal	Y	pointed	forward	N	claw	meat	Y	N	Y	2
mammal	Y	blunted	side	N	hoof	grass	Y	N	Y	2
bird	N	N	side	Y	claw	grain	N	N	N	1
bird	N	N	side	Y	web	fish	N	N	N	1
bird	N	N	side	Y	claw	grain	N	Y	N	1
bird	N	N	forward	Y	claw	meat	N	Y	N	1
other	N	pointed	forward	N	N	meat	N	N	N	1

Feature table T_A extraction for an attribute A from the generalized relation R

Input: A generalized relation R' consists of (i) an attribute A with distinct values $a_1,...,a_m$, m is the number of distinct values for A (ii) j other attributes $B_1,...,B_j$, j is the number of attributes in the relation R' except A (suppose different attributes have unique distinct values), and (iii) a special attribute, *vote*.

Output. The feature table T_A

Method.

1. The feature table T_A consists of $m+1$ rows and $l+1$ c
   total number of distinct values in all the attributes. Eac
   initialized to 0.

2. Each slot in T_A (except the last row) is filled by the follc

   for each row r in R' **do** {
   for each attribute B_j in R' **do**
   $T_A[r.A, r.B_j] := T_A[r.A, r.B_j] + r.vote$;
   $T_A[r.A, vote] := T_A[r.A, vote] + r.vote$; }

3. The last row p in T_A is filled by the following procedure:

   for each column s in T_A **do**
   for each row t (except the last row p) in T_A **do**
   $T_A[p, s] := T_A[p, s] + T_A[t, s]$;

Collection of “cars” information

.

Make_Model	fuel	disp	weight	cyl	power	turbo	comp	trans	mileage
Ford Escort	EFI	medium	876	6	high	yes	high	auto	medium
Dodge Shadow	EFI	medium	1100	6	high	no	medium	manu	medium
Ford Festiva	EFI	medium	1589	6	high	no	high	manu	medium
Chevrolet Corvette	EFI	medium	987	6	high	no	medium	manu	medium
Dodge Stealth	EFI	medium	1096	6	high	no	high	manu	medium
Ford Probe	EFI	medium	867	6	high	no	medium	manu	medium
Ford Mustang	EFI	medium	1197	6	high	no	high	manu	medium
Dodge Daytona	EFI	medium	798	6	high	yes	high	manu	high
Chrysler LeBaron	EFI	medium	1056	4	medium	no	medium	manu	medium
Dodge Sprite	EFI	medium	1557	6	high	no	medium	manu	low
Honda Civic	2-BBL	small	786	4	low	no	high	manu	high
Ford Escort	2-BBL	small	1098	4	low	no	high	manu	medium
Ford Tempo	2-BBL	small	1187	4	medium	no	high	auto	medium
Toyoto Corolla	EFI	small	1023	4	low	no	high	manu	high

Concept Hierarcy for “cars”.

attribute	concept	values
make_model	honda	civic, acura, ..., accord
	toyota	tercel, ..., camry
	mazda	mazda_323, mazda_626, ..., mazda 939
	japan (car)	honda, toyoto, ..., mazda
	ford	escort, probe, ..., taurus
	chevrolet	corvette, camaro, ..., corsica
	dodge	stealth, daytona, ..., dynasty
	usa (car)	ford, dodge, ..., chevrolet
	any (make-model)	japan (car), ..., usa (car)
	light	0, ..., 800
	heavy	801, ..., 1200
	medium	1201, ..., 1600
	any (weight)	light, medium, heavy

A generalized cars information system

Make_Model	fuel	disp	weight	cyl	power	turbo	comp	trans	mileage
USA	EFI	medium	medium	6	high	yes	high	auto	medium
USA	EFI	medium	medium	6	high	no	medium	manu	medium
USA	EFI	medium	heavy	6	high	no	high	manu	medium
USA	EFI	medium	medium	6	high	no	high	manu	medium
USA	EFI	medium	light	6	high	yes	high	manu	high
USA	EFI	medium	medium	4	medium	no	medium	manu	medium
USA	EFI	medium	heavy	6	high	no	medium	manu	low
Japan	2-BBL	small	light	4	low	no	high	manu	high
USA	2-BBL	small	medium	4	low	no	high	manu	medium
USA	2-BBL	small	medium	4	medium	no	high	auto	medium
Japan	EFI	small	medium	4	low	no	high	manu	high
Japan	EFI	medium	light	4	medium	no	medium	manu	high
Japan	EFI	small	medium	4	high	yes	high	manu	high
Japan	2-BBL	small	medium	4	low	no	medium	manu	high
USA	EFI	medium	medium	4	high	yes	medium	manu	medium
USA	EFI	medium	heavy	6	high	no	medium	auto	low
USA	EFI	medium	medium	6	high	no	medium	auto	medium
USA	EFI	medium	medium	4	high	no	medium	auto	medium
Japan	EFI	small	medium	4	medium	no	high	manu	high
USA	EFI	small	medium	4	medium	no	high	manu	high

A reduct of the generalized car information system

make_model	weight	power	comp	tran	mileage
USA	medium	high	high	auto	medium
USA	medium	high	medium	manu	medium
USA	heavy	high	high	manu	medium
USA	medium	high	high	manu	medium
USA	light	high	high	manu	high
USA	medium	medium	medium	manu	medium
USA	heavy	high	medium	manu	low
Japan	light	low	high	manu	high
USA	medium	low	high	manu	medium
USA	medium	medium	high	auto	medium
Japan	medium	low	high	manu	high
Japan	light	medium	medium	manu	high
Japan	medium	high	high	manu	high
Japan	medium	low	medium	manu	high
USA	heavy	high	medium	auto	low
USA	medium	high	medium	auto	medium
Japan	medium	medium	high	manu	high
USA	medium	medium	high	manu	high

The final reduced relation.

Make-model	compress	trans	milage
USA	HIGH	AUTO	MEDIUM
USA	MEDIUM	MANUAL	MEDIUM
USA	MEDIUM	AUTO	MEDIUM
USA	HIGH	MANUAL	HIGH
Japan	HIGH	MANUAL	HIGH
Japan	MEDIUM	MANUAL	HIGH

A set of maximally general rules

make_model	weight	power	comp	tran	mileage	supp
-	heavy	-	medium	-	low	2
USA	medium	high	-	-	medium	9
USA	medium	-	medium	-	medium	8
-	medium	-	-	auto	medium	4
USA	-	light	-	-	medium	1
-	heavy	-	high	-	medium	1
-	-	medium	high	manu	high	3
Japan	-	-	-	-	high	6
-	light	-	-	-	high	