Decision table

What is it for?

A decision table is used for modelling decisions with multiple outcomes based on one or more possible input conditions.

Please refer to the Decision Management guide for a guide on how to model logic.

What is it?

A decision table is a way to model logic. It is a compact graphical representation of a decision-making process that results in an action. When a decision table is complete and correct, decision tables provide transparency and explicitly present design decisions.

What does it do?

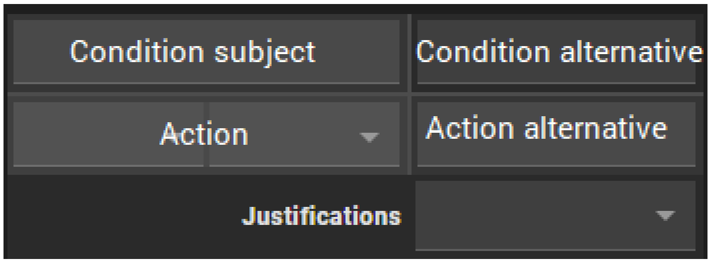
With a decision table you can derive one or more actions (decisions). To derive these actions we need one or more conditions (questions). The answers to these conditions are called condition alternatives and the possible decisions the action values. Typical knowledge tasks for which you use decision tables are assessing, classification and trying out. Examples of these are determining the right to child support or rent subsidies, and the need for a building permit or demolition permit.

Decision tables are particularly suitable when:

- Communication of knowledge is an important aspect;
- The knowledge is well structured;
- Reasoning about multiple instances is not required.

Advice on how to model logic in decision tables can be found here: Designing decision tables

Properties



Property	Description
Condition subject	Fill in the condition of the decision table. This value will be compared to the Condition alternatives on the same row.
Condition alternative	Fill in the condition alternatives.
Action	Choose any entity.attribute from the drop down list to create the action of the decision table.
Action alternative	Fill in the action alternatives.
Justification	The Justification contains an explanation behind the logic that is used in the decision table. Select the asset from the drop down list.

Condition subject

Most commonly this is an attribute, the value of which is going to be compared to the condition alternatives. Other common uses are a COLLECT statement to reason over multiple values at once. In general, this can be any expression. Special care should be taken when the Condition subject is a multi-valued expression, see Multivalued attributes in decision tables.

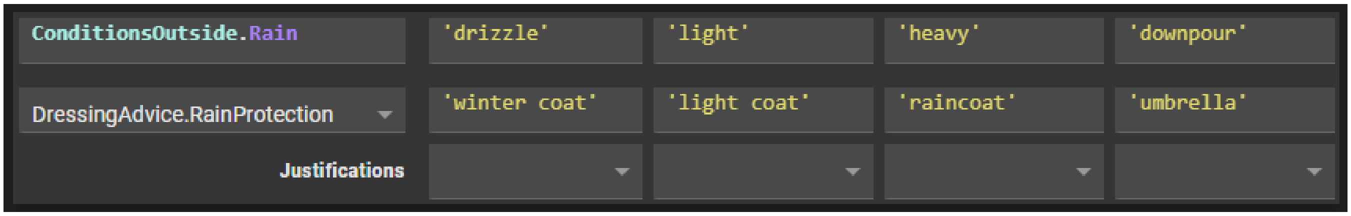
Condition alternatives

The condition alternatives can be either a single value (static or an expression), a multi-valued expression, or a construct consisting of multiple alternatives that are combined using AND and OR. There are also several Special condition alternatives, see below.

Single values

This the simplest case. Use either a static value of the appropriate type, an attribute, or any expression that results in a value of the appropriate type.

If you insert only a value in the Condition alternative, an implicit "=" is used to compare the Condition subject to this Condition alternative.



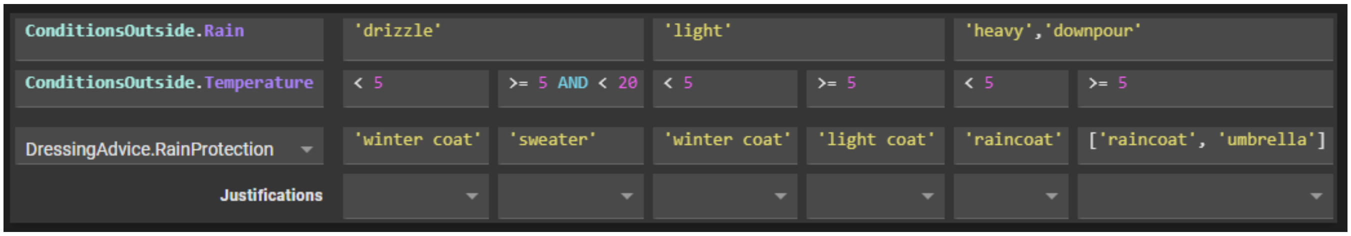
ConditionsOutside.Rain	'drizzle'	'light'	'heavy'	'downpour'
DressingAdvice.RainProtection	'winter coat'	'light coat'	'raincoat'	'umbrella'
Justifications				

You can also insert the comparison operators <, <=, >, >=, =, != in case the data type is integer, number or string.

It is also possible to use the boolean operators AND and OR to combine multiple values in one Condition alternative. Using AND between expressions means that both conditions should be met, using OR that either condition should be met.

Multi-valued expressions

When an expression is used in the Condition alternative that results in multiple values, the comparison acts just like any other where "=" is used. A static multi-valued expression is a list ["lowweight", "mediumweight"] (when only the list is present, the square brackets can be omitted), but it can also be a multi-valued attribute or a COLLECT statement.



ConditionsOutside.Rain	'drizzle'		'light'		'heavy','downpour'	
ConditionsOutside.Temperature	< 5	>= 5 AND < 20	< 5	>= 5	< 5	>= 5
DressingAdvice.RainProtection	'winter coat'	'sweater'	'winter coat'	'light coat'	'raincoat'	['raincoat', 'umbrella']
Justifications						

Special condition alternatives

Syntax	Meaning	Explanation
*	DON'T CARE	Use this condition alternative when the value of the corresponding condition (or the missing value, when the condition is unknown) is irrelevant for the conclusion (action alternative). Consequently, this DON'T CARE symbol * is always the single condition alternative below another one directly above it. In other words: it should never be used • in combination with other condition alternatives (implying that these values ARE NOT irrelevant, i.e. a contradiction), and • in the top row of a decision table (implying that the top condition is irrelevant, and that the entire top row is obsolete and can be removed)
?	UNKNOWN	Use this condition alternative to cover the situation that the value of the corresponding condition is unknown, but not irrelevant for the conclusion (otherwise use the DON'T CARE symbol *).
[]	ELSE	This covers all possible values that are not explicitly mentioned in other condition alternatives. UNKNOWN is not a value, so it is not covered by the ELSE. If you want to cover the UNKNOWN, you have to explicitly use the ? as well.

Decision tables are currently evaluated left-to-right. However, the ELSE [] will always be evaluated last, after the other explicitly mentioned values.

Actions

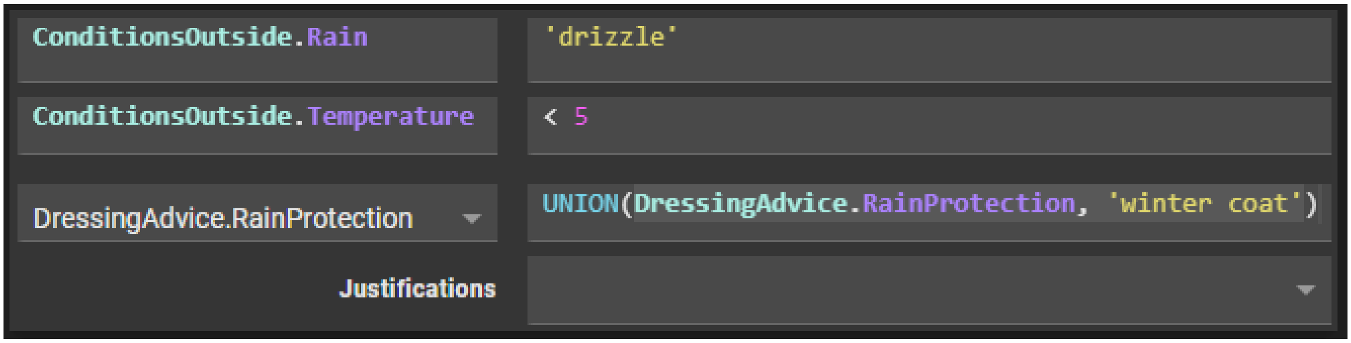
In most cases a decision table has only one action. It is possible to add Actions, so that multiple attributes will get a value.

Adding multiple actions to a decision table has to be used with caution, since it means that when the value of one of these attributes is derived by the inference engine, the value of all the other attributes will also be derived. This might violate the principle of "single responsibility", which says that any element in the model should only have one function.

Action alternatives

An Action alternative specifies the value that the attribute in the Action will get (when the conditions in the condition columns above are met). It should contain an expression of the appropriate data type. When the values of the Action attribute are restricted to a value list, the values of the Action alternative can be chosen from a pick list.

Note that it is possible to use the value of the Action attribute itself, in cases where you want to add a value to a multi-value attribute:



ConditionsOutside.Rain	'drizzle'
ConditionsOutside.Temperature	< 5
DressingAdvice.RainProtection	UNION(DressingAdvice.RainProtection, 'winter coat')
Justifications	

Special action alternative

Character	Meaning	Explanation
*	UNKNOWN	The Action attribute is derived as ? (unknown). Note that even if the attribute has a default value other than this decision table, the value ? derived by the decision table has a higher priority.

Add or remove conditions/actions/alternatives

Initially the decision table looks like this:

condition subject	*condition alternative*
action subject	*action value*

The decision table can be extended to contain multiple conditions, actions and/or alternatives. When you select a cell, you will see a control menu appear.

In a *condition* or *action*, you can use the + icons to insert a row above or below the selection. In a *condition alternative*, you can use the + icons to add a new column before or after the selection.

You also have the possibility to use the (shift+)'tab' on your keyboard to go from one input field to another.

Example:

If you select condition alternative CA 2.1 inserting a column to the right will result in this table:

When pressing the delete button in a condition or action cell, the whole row which that cell is in is removed. When you press the same button in a condition alternative of action value cell, the whole column is removed.

Adjust a column

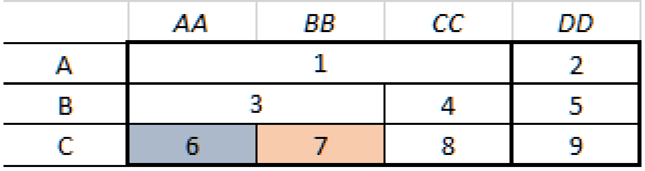
In a *condition alternative*, you can use the arrow icons to move a column from left to right or from right to left. These are restricted at the front and end of the row. The size of the column will be kept to his original width.

Note: You can not move a row up or down or drag and drop a field.

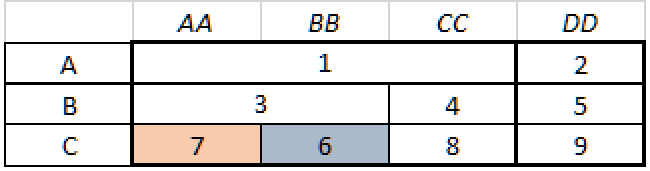
Examples:

In row C it is only possible to switch cell 6 and 7

AA-C gets BB-C



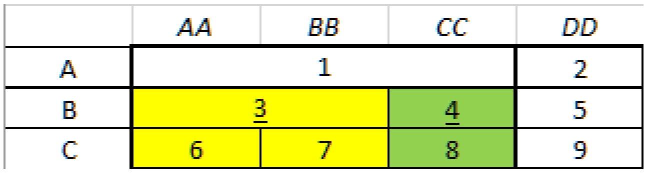
	AA	BB	CC	DD
A	1			2
B	3		4	5
C	6	7	8	9



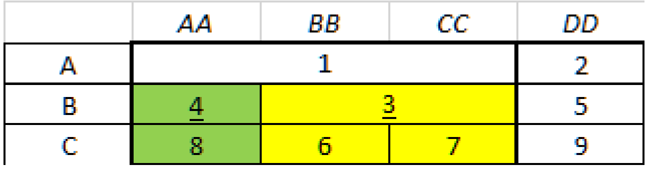
	AA	BB	CC	DD
A	1			2
B	3		4	5
C	7	6	8	9

If columns in row B switch, AA-BB-B will switch with CC-B. The underlying tree moves with the top node

AA-BB-B gets BB-CC-B



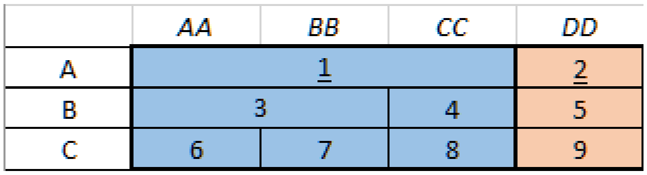
	AA	BB	CC	DD
A	1			2
B	3		4	5
C	6	7	8	9



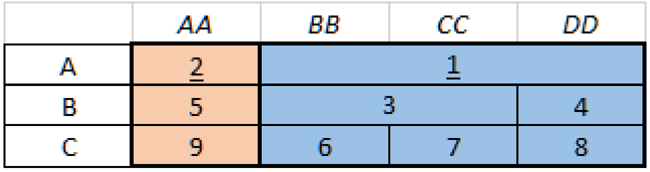
	AA	BB	CC	DD
A	1			2
B	4	3		5
C	8	6	7	9

If columns in row A switch then column AA-BB-CC will switch with DD. The underlying tree moves with the top node

AA-BB-CC gets BB-CC-DD



	AA	BB	CC	DD
A	1			2
B	3		4	5
C	6	7	8	9



	AA	BB	CC	DD
A	2	1		
B	5	3		4
C	9	6	7	8