

An Introductory Guide to Caspian

1. Introduction

This document deals with using the program Caspian. It gives a brief summary of how you use the program, and then deals in more detail with what happens when you run Caspian. It also has a list of error messages and some explanation for them.

Caspian creates a casebase from a case file written in the language CASL. The language itself is dealt with in the separate document "Creating a Case-Base using CASL".

We shall use the CHEF example (adapted from Riesbeck and Schank, *Inside Case-Based Reasoning*) of adapting a recipe for stir-frying chicken and green beans to a new recipe for stir-frying beef and broccoli to illustrate the steps involved.

2. Running the program

Caspian is invoked by the command `caspian <filename>`, where `<filename>` is the name of the case file written in CASL.

So to run the CHEF example, type `caspian chef` or `caspian chef -p` to get list expansion.

The screen should display an introductory message, followed by the following:

```
Reading case file...ok
Type checking...ok
Checking for doubly defined fields...ok
Checking for multiple fields...ok
Checking for doubly defined case names...ok
Checking for non-unique enumeration symbols...ok
Checking index block...ok
Checking modification rules...ok
Checking repair rules...ok
Building index structure...ok
```

These messages indicate that the case file has been read in and has passed various syntactic and semantic checks.

The sequence followed then is:

- 1) The user enters a case for comparison (the "user case").
- 2) The program performs an index search and finds a subset of cases that match all the index constraints.
- 3) If no cases match all the index constraints (very likely, as there are only a couple of cases in each example case base), then the system prompts you for different index values. Try *stir-fry* and *sweet and sour* to get a sensible result.
- 4) A case is selected from the subset.
- 5) Repairs are carried out on the solution part of the selected case.

Entering the user case

The program prompts the user for values for each of the problem fields. Some of the fields may be skipped over by pressing Return, or by typing semicolon and pressing Return, if the field type is a list. Fields which are used for indexing may not be skipped over.

So for the CHEF example the sequence is as follows:

Select value for cook_method

- 1) boil
- 2) bake
- 3) stir_fry

Enter choice (1 to 3): 3

Select value for goal:

- 1) include tofu
- 2) include cheese

Enter choice (1 to 2 or hit RETURN):

Select value for taste

- 1) hot
- 2) sweet
- 3) savoury
- 4) sweet and sour

Enter choice (1 to 4 or hot RETURN): 4

Select value for meat

- 1) beef
- 2) lamb
- 3) chicken

Enter choice (1 to 6 or hit RETURN): 1

Select value for vegetable

- 1) snow_peas
- 2) broccoli
- 3) green_beans

Enter choice (1 to 6 or hit RETURN): 2

The index search

The index constraints are taken from the field values input by the user. The program searches the casebase for the subset of cases that matches all the index constraints exactly. If there are no cases which match all the index constraints, a message to this effect is displayed and the user is prompted to enter new values for the index constraints. These may be made more general by specifying abstraction values or by simply specifying fewer constraints.

Selecting a Case from the Subset

After the index search has completed successfully, the case matcher is invoked to scan the subset of cases to find the one with the highest weight value. This is then selected and the repair rules are applied to it.

A warning about cooking broccoli will be displayed, followed by the repaired chicken and green beans recipe:

3. Error Messages

All errors that occur when reading/checking the case file are fatal and cause the program to exit with error code 1. Error messages beginning with "Huh?" are internal and cause the program to exit with error code 1.

Lexical Errors

Error - STRBUFSIZE not big enough

Error reading case file - lexeme is too long

These occur when a string or lexeme has been entered that is longer than 50 characters.

Error reading case file - unidentified lexeme

This is a syntax error at the lexical level.

Parsing Errors

Error reading case file - Invalid number range

This occurs when parsing a modification rule for a NUMBER field type. It means that the upper value of a range has a value which is actually less than that of the lower range.

Error reading case file - Result range is -5.0 to +5.0

This means that you have entered a number outside the -5 to 5 range required for a result value.

Error reading case file - syntax error

A syntax error has been found in the case file.

Case Definition Section Errors

Error -fieldname [symbol] in enumeration list

The fieldname of an enumeration definition has been included in the enumeration list.

Error - enumeration list contains [symbol] defined twice

A symbol list in an enumeration definition contains a symbol more than once.

Error - Fieldname [symbol] defined twice

There are two field definitions with the same fieldname.

Error - field [symbol] enumeration list needs at least two values

Although syntactically correct, an enumeration list should always contain at least two values.

Error - Fieldname [symbol] defined in both sections

The same fieldname is defined twice, once in the problem section and once in the solution section.

Index Definition Errors

Error - fieldname [symbol] not defined in problem part

A fieldname has been used which is not defined in the problem section of the case definition.

Modification Definition Section Errors

Error - abstraction value [symbol] included in similarity list

An abstraction symbol has been included in the symbol list of the same rule.

Error - fieldname [symbol] included in similarity list

The fieldname has been included in the symbol list of the same rule.

Error - enumeration list contains [symbol] defined twice

A symbol list in a modification rule contains a symbol more than once.

Error - doubly defined abstraction symbol [symbol]

Two modification rules for the same field have the same abstraction symbol.

Error - Fieldname [symbol] not defined in problem section

A modification rule is applied to a field not defined in the problem section of the case definition.

Case Instance Errors

Error - case name [symbol] is not unique

There are two or more cases with the given name.

Error - field [fieldname] used twice in case [casename]

A field has been used more than once in the same section of a case.

Error - Field [symbol] is not defined

Probably a spelling error in a fieldname.

Repair Rule Definition Section Errors

Error - Fieldname [symbol] used twice in repair rule

A fieldname has been used more than once in the replist part of a repair rule.

Error - fieldname [symbol] used twice in combination list

A fieldname has been used more than once in the comblist part of a repair rule

Error - fieldname [symbol] not defined in solution part.

A repair rule attempts to change a field not defined in the solution section.

User - Input Errors

Errors caused by user input should not cause the program to exit.

Error - symbol is too long Error - string is too long

The user entered a symbol/string more than 29 characters long. These errors can also occur when entering a list.

Error - Syntax Error in list

The user entered a list with a syntax error in it.

The user should re-enter the list.

Indexing related Errors

Error - indexed field [fieldname] value undefined in case [casename]

The case file contains a case where not all indexed fields are defined in the problem section.

Arithmetic related Errors

Error - Type mismatch - field [symbol] in arithmetical expression

The type of the field is not NUMBER.

Miscellaneous Errors

Error - Type mismatch error on field [symbol]

Error - Type mismatch with field [symbol]

A type error occurred assigning/comparing a field. May also occur in the index block if the field is not an enumerated type.

Error - Fieldname [symbol] not defined

Error - Fieldname: [symbol] does not exist

This is probably a spelling error in a fieldname.

Error - Enumeration symbol [symbol] not defined

Probably a spelling error in an enumeration symbol.