Category-partition Method

- Key idea
  - Method for creating test suites
  - Role of test engineer
    - Analyze the system specification
    - Write a series of formal test specifications
  - Automatic generator
    - Produces test frames

Steps

- Decompose the functional specification into functional units
  - Characteristics of functional units
    - They can be tested independently
    - Examples
      - A top-level user command
      - Or a function
- Decomposition may require several stages
- Similar to high-level decomposition done by software designers
  - May be reused, although independent decomposition is recommended

Steps

- Examine each functional unit
  - Identify parameters
    - Explicit input to the functional unit
  - Environmental conditions
    - Characteristics of the system's state
- Test Cases
  - Specific values of parameters
  - And environmental conditions

Steps

- "Test cases are chosen to maximize chances of finding errors"
- For each parameter & environmental condition
  - Find categories
    - Major property or characteristic
    - Examples
      - Browsers, Operating Systems, array size
    - For each category
      - Find choices
        - » Examples: (IE 5.0, IE 4.5, Netscape 7.0), (Windows NT, Linux), (100, 0, -1)

Steps

- Develop "Formal Test Specification" for each functional unit
  - List of categories
  - Lists of choices within each category
- Constraints
- Automatically produces a set of "test frames"
  - Consists of a set of choices

An Example Command

Command:
find

Syntax:
find <pattern> <file>

Function:
The find command is used to locate one or more instances of a given pattern in a text file. All lines in the file that contain the pattern are written to standard output. A line containing the pattern is written only once, regardless of the number of times the pattern occurs in it.

The pattern is any sequence of characters whose length does not exceed the maximum length of a line in the file. To include a blank in the pattern, the entire pattern must be enclosed in quotes ("). To include a quotation mark in the pattern, two quotes in a row ("") must be used.

Examples of Find Usage

Examples:

```
find john myfile
```
displays lines in the file **myfile** which contain **john**

```
find "john smith" myfile
```
displays lines in the file **myfile** which contain **john smith**

```
find "john"" smith" myfile
```
displays lines in the file **myfile** which contain **john" smith**

Analyzing the Specs

- Individual function that can be tested separately
- Two parameters
  - Pattern
  - File
- Pattern characteristics
  - From specs
    - Length
    - Enclosed in quotes or not
    - Embedded blanks or not
    - Embedded quotes or not
  - Not from specs
    - Quoted must have blanks?
    - Successive quotes?

Analyzing the Specs (2)

- File
  - Name is a parameter
    - File exists
    - Or not
  - File properties are environmental characteristics
    - Number of occurrences of pattern in file
    - Number of occurrences of pattern in a line
    - Maximum line length in a file

Test Specs - Parameters

```
Parameters:
    Pattern size:
        empty
        single character
        many character
        longer than any line in the file

    Quoting:
        pattern is quoted
        pattern is not quoted
        pattern is improperly quoted

    Embedded blanks:
        no embedded blank
        one embedded blank
        several embedded blanks

    Embedded quotes:
        no embedded quotes
        one embedded quote
        several embedded quotes

    File name:
        good file name
        no file with this name
        omitted
```

Test Specs - Environment

```
Environments:
    Number of occurrences of pattern in file:
        none
        exactly one
        more than one

    Pattern occurrences on target line:
    #  assumes line contains the pattern
        one
        more than one
```

Number of Test Frames

- 1944

Contradictory Requirements

- Can we even generate such a test case?

```
Pattern size : empty
Quoting : pattern is quoted
Embedded blanks : several embedded blanks
Embedded quotes : no embedded quotes
File name : good file name
Number of occurrences of pattern in file : none
Pattern occurrences on target line : one
```

Constraints

- Properties
  - [property A, B, …]
  - A and B are property names
  - E.g., [property Empty]
- Selector expression
  - [if A]
  - E.g., [if Empty]

Adding Constraints

```
Parameters:
    Pattern size:
        empty                                   [property Empty]
        single character                        [property NonEmpty]
        many character                          [property NonEmpty]
        longer than any line in the file        [property NonEmpty]

    Quoting:
        pattern is quoted                       [property Quoted]
        pattern is not quoted                   [if NonEmpty]
        pattern is improperly quoted            [if NonEmpty]

    Embedded blanks:
        no embedded blank                       [if NonEmpty]
        one embedded blank                      [if NonEmpty and Quoted]
        several embedded blanks                 [if NonEmpty and Quoted]

    Embedded quotes:
        no embedded quotes                      [if NonEmpty]
        one embedded quote                      [if NonEmpty]
        several embedded quotes                 [if NonEmpty]

    File name:
        good file name
        no file with this name
        omitted

Environments:
    Number of occurrences of pattern in file:
        none                                    [if NonEmpty]
        exactly one                             [if NonEmpty] [property Match]
        more than one                           [if NonEmpty] [property Match]

    Pattern occurrences on target line:
        #  assumes line contains the pattern
        one                                     [if Match]
        more than one                           [if Match]
```

Number of Test Frames

- 678
- Can we reduce them?

Adding [error] and [single]

```
Parameters:
    Pattern size:
        empty                                   [property Empty]
        single character                        [property NonEmpty]
        many character                          [property NonEmpty]
        longer than any line in the file        [error]

    Quoting:
        pattern is quoted                       [property Quoted]
        pattern is not quoted                   [if NonEmpty]
        pattern is improperly quoted            [error]

    Embedded blanks:
        no embedded blank                       [if NonEmpty]
        one embedded blank                      [if NonEmpty and Quoted]
        several embedded blanks                 [if NonEmpty and Quoted]

    Embedded quotes:
        no embedded quotes                      [if NonEmpty]
        one embedded quote                      [if NonEmpty]
        several embedded quotes                 [if NonEmpty] [single]

    File name:
        good file name
        no file with this name                  [error]
        omitted                                 [error]

Environments:
    Number of occurrences of pattern in file:
        none                                    [if NonEmpty] [single]
        exactly one                             [if NonEmpty] [property Match]
        more than one                           [if NonEmpty] [property Match]

    Pattern occurrences on target line:
    #  assumes line contains the pattern
        one                                     [if Match]
        more than one                           [if Match] [single]
```

Number of Test Frames

- [error]
  - 125
- [single]
  - 40

Generating Test Cases

- Use a constraint solver
- Choose specific values that satisfy the constraints