nice interface for purposes of the course, as long as you are able to show us with your write-up which are the interesting tasks that your system can perform. (If you want to use the project for a larger purpose, of course you will need an interface.)

- Since the aim of the project is to exploit Classic capabilities as a knowledge representation language and to identify its limitations, it is very important that, every time you use a LISP function to perform an inference or a TEST-C in the definition of a concept, you describe why you could not reach your goal using Classic constructs only.

  In general, it is important that you carefully describe all the limitations (if any) that Classic expressive power imposed on reasoning tasks you tried to implement.

An example of write-up of a project for the representation of the Real-Estate domain can be found in

```
/usr/local/ai-tool/classic/kr/kr-lab/real-estate.ps
```

You still must hand in the files with your code, *properly* commented, but do not expect that we will understand from the code things you did not make clear in the write-up.

Guidelines for Writing up a Project

The write-up is a fundamental component of your project. Your write-up must include the following information.

- A general description of the domain. This part is particularly important if the domain you chose is at all specialized or requires any expertise. In writing this part you should bear in mind that we cannot judge your representation of the domain if we do not understand the domain itself.

- A description of the Classic hierarchy. A log file printing out the output of (cl-concept-descendants @classic-thing) is *not* a description of the hierarchy. You should describe in English the concepts and the way they are organized in the hierarchy, explaining why you chose a certain representation when you think that there might be alternative acceptable representations or when you believe that the explanation is not obvious given the description of the domain. For the most important or interesting concepts, the best approach would be to insert the Classic representation of the concept after its English description. A graphical representation of the KB usually helps to clarify this part of the presentation. But since no graphic tool is available (at least up to now), such graphical representation is optional.

- A description of the tasks and salient inferences that your knowledge base performs. This part usually becomes more clear if the English explanation of the inference is followed by the actual Classic trace. If you are writing your document in LaTeX, you can use the following LaTeX construct to insert extracts of the Classic trace and of your code in the document.

  ```
  \begin{verbatim}

  code or trace you want to insert
  ```

  The same criterion applies to LISP functions that you believe can clarify how an inference has been performed. In the write-up you can just describe what the function does, or insert the code itself if you think it is important. Interface functions are usually *not* important. As a matter of fact you should not spend much effort in implementating a