APPENDIX G

Instructions for Recompiling the PDP Programs

Complete source code for all the programs described in this book is provided in the archive *src.arc*. If you have Version 3.0 (or higher) of the Microsoft C compiler, you should be able to modify the programs and recompile them on your PC. If you have a UNIX system (Berkeley 4.2 or higher) with the standard UNIX C compiler, you should also be able to recompile the programs to run on that system. We first provide a general inventory of the components of the PDP software and of the dependencies among these components. Then we describe the procedure for recompiling for the PC with Microsoft C. Following this we briefly describe how to set up the PDP software on UNIX systems. You are free to try to use other C compilers, but with others you are completely on your own. There is every reason to expect that some tinkering will be required to recompile the software with non-UNIX C compilers other than Microsoft C.

Components of the PDP Software

The software includes seven executable programs: **aa, bp, cl, cs, ia, iac,** and **pa**. For each of these programs, there is a source file with the same name (e.g., *aa.c*). In addition, there are certain other source files that several programs share. The object files that all programs share are grouped into one library file called *libpc.a*. This library file is made up of the compiled versions of the routines from the files *command.c, display.c, general.c, io.c, main.c, patterns.c, template.c,* and *variable.c*. The software also includes the two utility programs: **plot** and **colex**; each of these is constructed from a single corresponding *.c* file.

If you change any source file, you should recompile and relink all object and executable files that depend upon that source file. The following dependency list shows which executables depend upon which source files. Where *libpc.a* is shown, all eight source files in the library are included in the dependency.

aa:	*aa.c, libpc.a*
bp:	*bp.c, weights.c, libpc.a*
cl:	*cl.c, libpc.a*
cs:	*cs.c, weights.c, libpc.a*
ia:	*ia.c, iaaux.c, iatop.c, libpc.a*
iac:	*iac.c, weights.c*
pa:	*pa.c, weights.c, libpc.a*
plot:	*plot.c*
colex:	*colex.c*

Note also that there are many header files (with names ending in *.h*) in the PDP software package. These files often contain declarations that are used in several different modules. This is particularly true for the header files associated with the modules in *libpc.a*. If one of these files is modified, it is prudent to recompile all modules that include this file. The following list indicates which *.h* files are included in each *.c* file:

general.h:	*display.h*
aa.c:	*general.h, aa.h, variable.h, patterns.h, command.h*
bp.c:	*general.h, bp.h, variable.h, weights.h, patterns.h, command.h*
cl.c:	*general.h, variable.h, patterns.h, command.h, cl.h*
command.c:	*general.h, io.h, command.h*
cs.c:	*general.h, cs.h, variable.h, command.h, patterns.h, weights.h*
display.c:	*general.h, io.h, variable.h, template.h, weights.h, command.h*
general.c:	*general.h, command.h, variable.h*
ia.c:	*ia.h, io.h, general.h*
iaaux.c:	*ia.h*
iac.c:	*general.h, iac.h, variable.h, command.h, weights.h, patterns.h*
iatop.c:	*general.h, cs.h, variable.h, command.h, ia.h*
io.c:	*io.h*
main.c:	*general.h, variable.h, command.h, patterns.h*
pa.c:	*general.h, pa.h, variable.h, weights.h, patterns.h, command.h*
patterns.c:	*general.h, command.h, variable.h, patterns.h*
template.c:	*general.h, command.h, variable.h, display.h, template.h*

variable.c:	*general.h, variable.h, command.h, patterns.h, weights.h*
weights.c:	*general.h, command.h, weights.h, variable.h*

Note that all files that include *general.h* also implicitly include *display.h*.

To Recompile for a PC Using Microsoft C

Three batch files are provided in the *src* directory to aid in compiling and linking: *compile.bat, makelib.bat,* and *pdplink.bat*. These files can be executed as though they were programs.

The *compile* batch file includes the command to compile a source file (*.c*) into an object file (*.obj*). The first step in creating an executable (*.exe*) file is to recompile all of the source files upon which it depends, including the files in *libpc.a*. To recompile all the source files, execute the following command in the directory that contains the files:

compile all

This will produce a *.obj* file for each *.c* file in that directory. To compile only one source file, use its name instead of *all* when giving the *compile* command. Thus, to compile *aa.c* you would enter

compile aa.c

This will produce a file called *aa.obj* in that directory. If there are errors in your source code, the compiler may abort the command file and display the error messages.

Once all of the necessary object files are created, the *libpc.a* file can be built. The command file for building the library is executed as follows:

makelib

This will create a file called *libpc.a* in the current directory from the eight object files, which should be in the same directory. Whenever you recompile any of the eight programs in *libpc.a*, you should use the *makelib* command again to update the library. (The *makelib* command requires all the object files to be present, so it is best to keep these files around while you are actively involved in modifying the programs. Once you stop making changes, you can delete the *.obj* files to save space.)

The final stage in compiling is linking. The command file *pdplink.bat* will link the necessary files to create each of the executables. To link the object files for a particular program, enter *pdplink* with the program name as argument. Thus,

pdplink bp

will link the files *bp.obj* and *weights.obj* with the library *libpc.a* to create the executable *bp.exe*. The *pdplink* command will also work with **plot** or **colex**. To create executables for all seven PDP simulation programs, use the following command:

pdplink all

Note that *pdplink all* does not link **colex** and **plot**; these must be linked individually.

If you wish to recompile and relink all of the programs at once, use the following three commands:

compile all
makelib
pdplink all

Once you have created new executables, you will want to move these files into the appropriate working directories, to be used with the relevant *.tem* and *.str* files. The MS-DOS copy utility can be used to do this.

Instructions for Setting Up the PDP Software on UNIX Systems

For UNIX systems, we suggest that you set up a parent directory system for the PDP software and copy the extracted contents of each of the *.arc* files into a separate subdirectory of the parent, giving the subdirectory the same name as the archive. For example, if your parent directory were called */usr/yourname/pdp*, you would put the contents of *aa.arc* into a subdirectory of this directory called */usr/yourname/pdp/aa*. The only files that you will not want to include in this directory system are the *.exe* files, since these will only run on PCs. You would also create a subdirectory called */usr/yourname/pdp/src* containing the source files and other materials necessary to recompile the package from the *src.arc* file.

Once the directories have been set up, you will want to change directories to the *src* subdirectory. It is an easy matter to recompile all the programs because we have supplied a *makefile*. This file is used by the UNIX **make** program to manage the PDP software. To compile all of the PDP simulation programs, you need only execute the following command:

make

To compile a single program, simply give *make* the name of that program as an argument. For example, to recompile the **aa** program, enter

make aa

This form also works with the **plot** and **colex** programs; they are not updated if **make** is executed with no arguments.

In either case, **make** will check the *makefile* to see which source files need to be recompiled and will recompile them. It will update *libpc.a* if necessary. And it will link the necessary object modules together to create the necessary executable files. The supplied makefile places the seven PDP executables in directories that are on the same level as the source directory and have the same name as the executable. For example, if the *src* directory is */usr/yourname/pdp/src* then the **aa** executable would be placed in the directory */usr/yourname/pdp/aa*. If you have set up subdirectories for each program as suggested above, this will all work fine. If you have chosen to organize the directories differently, the *makefile* can be modified to change where each program is placed. For each program there is a variable that specifies the destination directory for the executable version of the program. The names of these variables are uppercase and consist of the program name followed by *DEST*. Thus for **aa**, there is a line in the *makefile* that looks like this:

```
AADEST = ../aa/
```

The path name to the right of the equal sign can be replaced by any other valid UNIX path name. Once it is, **aa** will be stored in the directory specified by the path. Thus

```
AADEST = /usr/foo/pc/bin/
```

would cause **make** to put the **aa** executable in the directory */usr/foo/pc/bin*.

References

Adams, M. J. (1979). Models of word recognition. *Cognitive Psychology, 11,* 133-176.

Anderson, J. A. (1977). Neural models with cognitive implications. In D. LaBerge & S. J. Samuels (Eds.), *Basic processes in reading perception and comprehension* (pp. 27-90). Hillsdale, NJ: Erlbaum.

Anderson, J. A. (1983). Cognitive and psychological computation with neural models. *IEEE Transactions on Systems, Man, and Cybernetics*, *13*, 799-815.

Anderson, J. A., Silverstein, J. W., Ritz, S. A., & Jones, R. S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review*, *84*, 413-451.

Bagley, W. C. (1900). The apperception of the spoken sentence: A study in the psychology of language. *American Journal of Psychology*, *12*, 80-130.

Baron, J., & Thurston, I. (1973). An analysis of the word-superiority effect. *Cognitive Psychology, 4,* 207-228.

Blake, A. (1983). The least disturbance principle and weak constraints. *Pattern Recognition Letters*, *1*, 393-399.

Broadbent, D. E., & Gregory, M. (1968). Visual perception of words differing in letter digram frequency. *Journal of Verbal Learning and Verbal Behavior*, *7*, 569-571.

Carr, T. H., Davidson, B. J., & Hawkins, H. L. (1978). Perceptual flexibility in word recognition: Strategies affect orthographic computation but not lexical access. *Journal of Experimental Psychology: Human Perception and Performance,* 4, 674-690.

Cattell, J. M. (1886). The time taken up by cerebral operations. *Mind, 11,* 220-242.

Feldman, J. A. (1981). A connectionist model of visual memory. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel models of associative memory* (pp. 49-81). Hillsdale, NJ: Erlbaum.

Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, *20*, 121-136.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721-741.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: Part I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, *23*, 121-134.

Grossberg, S. (1978). A theory of visual coding, memory, and development. In E. L. J. Leeuwenberg & H. F. J. M. Buffart (Eds.), *Formal theories of visual perception*. New York: Wiley.

Grossberg, S. (1980). How does the brain build a cognitive code? *Psychological Review*, *87*, 1-51.

Hebb, D. O. (1949). *The organization of behavior*. New York: Wiley.

Hinton, G. E. (1977). *Relaxation and its role in vision.* Unpublished doctoral dissertation, University of Edinburgh.

Hinton, G. E., & Anderson, J. A. (Eds.). (1981). *Parallel models of associative memory*. Hillsdale, NJ: Erlbaum.

Hinton, G. E., & Sejnowski, T. J. (1983). Optimal perceptual inference. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 448-453.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, *79*, 2554-2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences, USA*, *81*, 3088-3092.

James, W. (1890). *Principles of psychology* (Vol. 1). New York: Holt.

Johnston, J. C. (1978). A test of the sophisticated guessing theory of word perception. *Cognitive Psychology, 10,* 123-153.

Johnston, J. C. (1980). Experimental tests of a hierarchical model of word identification. *Journal of Verbal Learning and Verbal Behavior, 19,* 503-524.

Johnston, J. C., & McClelland, J. L. (1973). Visual factors in word perception. *Perception & Psychophysics, 14,* 365-370.

Johnston, J. C., & McClelland, J. L. (1974). Perception of letters in words: Seek not and ye shall find. *Science, 184,* 1192-1194.

Johnston, J. C., & McClelland, J. L. (1980). Experimental tests of a hierarchical model of word identification. *Journal of Verbal Learning and Verbal Behavior, 19,* 503-524.

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the Eighth Annual Meeting of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Kernighan, B. W., & Ritchie, D. M. (1978). *The C Programming Language.* Englewood Cliffs, NJ: Prentice-Hall.

Kohonen, T. (1977). *Associative memory: A system theoretical approach.* New York: Springer.

Kucera, H., & Francis, W. (1967). *Computational analysis of present-day American English*. Providence, RI: Brown University Press.

Levin, J. A. (1976). *Proteus: An activation framework for cognitive process models* (Tech. Rep. No. ISI/WP-2). Marina del Rey, CA: University of Southern California, Information Sciences Institute.

Luce, R. D. (1959). *Individual choice behavior.* New York: Wiley.

Manelis, L. (1974). The effect of meaningfulness in tachistoscopic word perception. *Perception & Psychophysics, 16,* 182-192.

Massaro, D. W. (1973). Perception of letters, words, and nonwords. *Journal of Experimental Psychology, 13,* 45-48.

Massaro, D. W., & Klitzke, D. (1979). The role of lateral masking and orthographic structure in letter and word recognition. *Acta Psychologica, 43,* 413-426.

McClelland, J. L. (1976). Preliminary letter identification in the perception of words and nonwords. *Journal of Experimental Psychology: Human Perception and Performance,* 2, 80-91.

McClelland, J. L. (1979). On the time-relas of mental processes: An examination of systems of processes in cascade. *Psychological Review*, *86*, 287-330.

McClelland, J. L. (1981). Retrieving general and specific information from stored knowledge of specifics. *Proceedings of the Third Annual Meeting of the Cognitive Science Society*, 170-172.

McClelland, J. L., & Johnston, J. C. (1977). The role of familiar units in perception of words and nonwords. *Perception & Psychophysics, 22,* 249-261.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, *88*, 375-407.

McClelland, J. L., & Rumelhart, D. E. (1985). Distributed memory and the representation of general and specific information. *Journal of Experimental Psychology: General*, *114*, 159-188.

McClelland, J. L., Rumelhart, D. E., & the PDP Research Group. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 2. Psychological and biological models.* Cambridge, MA: MIT Press/Bradford Books.

Minsky, M., & Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Morton, J. (1969). Interaction of information in word recognition. *Psychological Review, 76,* 165-178.

Norman, D. A., & Bobrow, D. G. (1975). On data-limited and resource-limited processes. *Cognitive Psychology*, *7*, 44-64.

Pillsbury, W. B. (1897). A study in apperception. *American Journal of Psychology, 8,* 315-393.

Pinker, S., & Prince, A. (1987). *On language and connectionism: Analysis of a parallel distributed processing model of language acquisition* (Occasional Paper 33). Cambridge: Massachusetts Institute of Technology, Center for Cognitive Science.

Reicher, G. M. (1969). Perceptual recognition as a function of meaningfulness of stimulus material. *Journal of Experimental Psychology, 81,* 274-280.

Riley, M. S., & Smolensky, P. (1984). A parallel model of (sequential) problem solving. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society.*

Rosenblatt, F. (1959). Two theorems of statistical separability in the perceptron. In *Mechanisation of thought processes: Proceedings of a symposium held at the National Physical Laboratory, November 1958. Vol. 1* (pp. 421-456). London: HM Stationery Office.

Rosenblatt, F. (1962). *Principles of neurodynamics*. New York: Spartan.

Rumelhart, D. E. (1977). Toward an interactive model of reading. In S. Dornic (Ed.), *Attention & Performance VI.* Hillsdale, NJ: Erlbaum.

Rumelhart, D. E., & McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extensions of the model. *Psychological Review*, *89*, 60-94.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1. Foundations.* Cambridge, MA: MIT Press/Bradford Books.

Rumelhart, D. E., & Ortony, A. (1977). The representation of knowledge in memory. In R. C. Anderson, R. J. Spiro, & W. E. Montague (Eds.), *Schooling and the acquisition of knowledge* (pp. 99-135). Hillsdale, NJ: Erlbaum.

Rumelhart, D. E., & Siple, P. (1974). Process of recognizing tachistoscopically presented words. *Psychological Review, 81,* 99-118.

Rumelhart, D. E., & Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science*, *9*, 75-112.

Selfridge, O. G. (1955). Pattern recognition in modern computers. *Proceedings of the Western Joint Computer Conference.*

Smolensky, P. (1983). Schema selection and stochastic inference in modular environments. *Proceedings of the National Conference on Artificial Intelligence AAAI-83*, 109-113.

Spoehr, K., & Smith, E. (1975). The role of orthographic and phonotactic rules in perceiving letter patterns. *Journal of Experimental Psychology: Human Perception and Performance*, *1*, 21-34.

Turvey, M. (1973). On peripheral and central processes in vision: Inferences from an information processing analysis of masking with patterned stimuli. *Psychological Review*, *80*, 1-52.

von der Malsberg, C. (1973). Self-organizing of orientation sensitive cells in the striate cortex. *Kybernetik*, *14*, 85-100.

Weisstein, N., Ozog, G., & Szoc, R. (1975). A comparison and elaboration of two models of metacontrast. *Psychological Review*, *82*, 325-343.

Wheeler, D. D. (1970). Processes in word recognition. *Cognitive Psychology, 1,* 59-85.

Whittlesea, B. W. A. (1983). *Representation and generalization of concepts: The abstractive and episodic perspectives evaluated.* Unpublished doctoral dissertation, MacMaster University.

Widrow, G., & Hoff, M. E. (1960). Adaptive switching circuits. *Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record*, *Part 4*, 96-104.

Index