IT 11 031
Examensarbete 15 hp
Juni 2011

Input Format Design and Translator Development for NJOY

C. Emil Hessman

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Input Format Design and Translator Development for NJOY

C. Emil Hessman

The NJOY Nuclear Data Processing System is a software system used for nuclear data management [1]. In particular, it is used to convert evaluated nuclear data for materials stored in Evaluated Nuclear Data Files (ENDF) [2] into different formats, as well as performing operations on the data.
NJOY is widely used within nuclear data research, and as such, it is important that the system has a user friendly interface. The NJOY input instructions [4] is a non-interactive user interface used for specifying jobs to be run by NJOY. The input instructions are complex and hard to read compared to e.g. a high-level programming language. Working with a large and complex job easily becomes a daunting and error-prone task. Accordingly, there is a need for an improved input format.
In this thesis, a new input format has been designed. In order to make the new input format useable with NJOY, a translator which is able to translate the new input format into the original NJOY input instructions has also been implemented. The results have been verified by a small set of tests.

Handledare: Gustav Wallin
Ämnesgranskare: Henrik Sjöstrand
Examinator: Anders Jansson
IT 11 031
Tryckt av: Reprocentralen ITC

Contents

1 Introduction

1.1 Background

Usability of software systems is important. The usability of a software system is determined, among other things, by its user interface. The user interface of a software system should provide means of interaction between the users and the system such that the desired result can be produced in an easy, elegant, and efficient fashion.

The NJOY Nuclear Data Processing System [1] is a software system used for nuclear data management. In particular, it is used to convert evaluated nuclear data for materials stored in Evaluated Nuclear Data Files (ENDF) [2] into different formats, as well as performing operations on the data. ENDF is a file format used for storing nuclear data which has been produced through an evaluation process.

NJOY is widely used within nuclear data research, and as such, it is important that the system has a user friendly interface. NJOY is currently being used within the MACRO project [3] at the Division of Applied Nuclear Physics, at the Department of Physics and Astronomy at Uppsala University. MACRO is a project that aims at linking reactor parameter uncertainties to uncertainties in nuclear cross sections and nuclear model parameters. This will be done using Monte Carlo-methods, and will require nuclear data processing on a large scale. It has been apparent to the researchers within the project that the non-interactive user interface used for specifying NJOY jobs is not as user friendly as one would like.

1.2 Problem Description

The NJOY input instructions [1, 4] are used for specifying jobs to be run by NJOY. It is a non-interactive user interface in the sense that the entire job needs to be specified prior to feeding the job to NJOY. No further input will be given when the processing of the job has started.

The NJOY input instructions are complex and hard to read compared to e.g. a high-level programming language. For example, algorithm 1 on the following page is a *short* and *simple* NJOY job which illustrates what the input instructions look like.

Algorithm 1 NJOY Test Problem 14

```
1  acer
2  20 21 0 31 32
3  1 0 1/
4  'proton + 7-n-14 apt la150 njoy99 mcnpx'/
5  725 0./
6  /
7  /
8  acer
9  0 31 33 34 35
10 7 1 2/
11 'proton + 7-n-14 apt la150 njoy99 mcnpx'/
12 viewr
13 33 36/
14 stop
```

Without consulting the documentation, one might guess that line 4 and 11 are some kind of descriptive titles, which is correct. One might also guess that line 14 terminates the program. However, it is not obvious that line 2 denotes input and output files (each number indicates a specific file) that the system will operate on. It is also hard to deduce that the first number on line 5 denotes the material to be processed, and that the second number denotes the desired temperature in kelvin.

The input instructions can be annotated with descriptive comments, but even then, working with a large and complex job easily becomes a daunting and error-prone task.

1.3 Objective

The NJOY input instructions is not an optimal input format. Therefore, the scope of this thesis has been to design and implement a more user friendly, and readable input format. The design of the new input format could be based on some commonly known existing format that is fitting to the task. The basis could for example be a programming language.

In order to make the new input format useable with NJOY, it has to be translated into the original NJOY input instructions. As such, the scope of this work also included developing an accompanying translator for the new input format.

2 NJOY Input Instructions

The NJOY input instructions is described in reference [1, 4]. A brief summary of the general structure of the input instructions is provided in section 2.1 as a convenience for the reader.

2.1 General Structure

NJOY is composed by a set of modules where each module performs a specific task. Each module has its separate input specification, which defines the expected input instructions for the module.

An NJOY job is an ordered sequence of modules, where the order denotes the execution order. The name of a module is used to denote the start of the corresponding module and its specific input instructions. The input instructions for a module is composed by an ordered sequence of *cards*. A *card* is a line with an ordered sequence of values separated by spaces (or commas). A card may be terminated with a slash character to denote the end of the card, but it is not required. Algorithm 2 illustrates the general structure of an NJOY job.

Algorithm 2 General structure of an NJOY job

```
1  module_name
2  value value ... value
3  value value ... value
4  ...
5  value value ... value
6  module_name
7  value value ... value
8  value value/
9  ...
10 value value ... value
11
12 ...
13
14 module_name
15 value value ... value
16 /
17 ...
18 value value ... value
19 stop
```

Line 1 denotes the start of the first module and its specific input instructions. Line 2 denotes the first card for the module declared on line 1. Line 3 through 5 denotes input instructions for successive cards that also belongs to the module declared on line 1. Line 6 through 10 denotes the declaration of another module and its corresponding cards. Successive module definitions, lines 11 through 18, may follow. Line 8 and 16 shows cards that have been terminated with the

slash character, denoting that no more values has been defined for the cards. An NJOY job is terminated by the `stop` instruction as indicated by line 19.

The expected cards in a module and the expected values within a card depends on the specific input specification for the module. The expected type of the values also depends on the input specification for the specific card and module. There are three kinds of types that the values in a card may be defined as: floating-point numbers, natural numbers (integers), and character strings. Character strings are generally required to be terminated by a slash character.

A card may have default values. A default value is a value that does not have to be defined in an NJOY job. If the value is not defined, then a default value will be set by NJOY. Values that may be defaulted are always defined at the end of the card. Hence, the values in a card are organized such that values that must be defined are always defined prior to values that may be defaulted. Default values will be used when a card is terminated by a slash character. For example, according to reference [4], card 3 in the `reconr` module is composed by three values. The last two values have default values. Declaring the card as

```
value/
```

will set the first value to `value` while the last two values will be set to their default values internally by NJOY since no more values were defined in the card.

To conclude, the NJOY input instructions are powerful. It is possible to construct a complex NJOY job by declaring a long chain of modules with their specific input instructions. Or, an NJOY job may be as simple as a single `stop` instruction, which just terminates the job.

3 Methodology

3.1 Introduction

The NJOY input instructions had to be understood in order to design the new input format. Each module in the NJOY software system, as described in reference [1, 4], was analyzed separately such that a general structure and common language features could be extracted and used for further analysis.

The NJOY modules are listed in table 1. The modules have been prioritized with a number. The number indicates the importance level of the module to the MACRO project [3]. A low number indicates high priority, e.g. a module with a low number had to be implemented before a module with a higher number.

NJOY Module	**Priority**
acer	1
broadr	1
ccccr	3
covr	1
dtfr	3
errorr	1
gaminr	3
gaspr	3
groupr	1
heatr	1
leapr	3
matxsr	3
mixr	3
moder	1
plotr	2
powr	3
purr	3
reconr	1
resxsr	3
thermr	1
unresr	3
viewr	2
wimsr	3

Table 1: Implementation priority of the NJOY modules

As stated in reference [5], a translator (*compiler*) is a program that can read a program in one language and translate it into an equivalent program in another language. In the following subsections, principles and techniques for constructing a translator presented in reference [5], is described.

3.2 Designing the New Input Format

The syntax definition of the new input format was specified in a notation called context-free grammar [6]. A context-free grammar is a convenient method of specifying the syntax of a programming language. For instance, the assignment (*declaration*) of an identifier can have the form

```
material = 9237
```

which can be expressed in a context-free grammar as the production

```
assignment ::= l_value "=" r_value
```

where `l_value` and `r_value` are other productions expressing the structure of the left and right hand side of the assignment, respectively.

3.3 Building the Translator

In reference [5], the translation process is described as a sequence of phases. Each phase inspects and transforms a representation of the source program to another. Phases such as lexical analysis, syntax analysis, and semantic analysis has been used throughout this work and is described in section 3.3.1, 3.3.2, and 3.3.3 respectively.

The translator, which is supposed to translate the input format into NJOY input instructions, was partly constructed using a lexical-analyzer generator [7] and a parser generator [8]. The translator was written in the Python programming language [9], in a Unix-like environment.

3.3.1 Lexical Analysis

Lexical analysis is the process of dividing the source program into sequences of characters, called tokens [10]. Each token describes a group of characters in the source program as an abstract type.

For example, the identifier `material`, the assignment character, `=`, and the integer `9237` could be represented as tokens of the form

`<IDENTIFIER, material>`,

`<ASSIGNMENT, =>`, and

`<INTEGER, 9237>`

PLY Lex [11] was used to generate a lexical analyzer (*lexer*) for the input format. The method of identifying the tokens was implemented by using the notation of regular expressions [12] in PLY Lex.

3.3.2 Syntax Analysis

Syntax analysis is the process of creating a tree-like representation, an abstract syntax tree, composed of the tokens generated by the lexical analyzer [13]. The syntax tree is used to describe the grammatical structure of the source program.

PLY Yacc [11] was used to generate a syntax analyzer (*parser*) for the grammar definition of the input format. The method of building the syntax tree was implemented by using the facilities provided by the PLY tools.

3.3.3 Semantic Analysis

Semantic analysis is the process of checking the syntax tree for errors that have to do with the *meaning* of the program [14].

For example, according to reference [4], card 1, 2 and 3 in the `acer` module must always be defined, and they must be defined in sequential order. The translator should report an error if these rules are violated; such as when card 1 has not been defined or when card 3 has been defined prior to card 2.

Type checking is another important part of the semantic analysis where the translator checks that each operator has valid operands.

For example, the identifier `hk`, in card 3 module `acer`, is used to denote a descriptive character string. According to reference [4], `hk` must be declared as a character string and must not exceed 70 characters in length. The translator should report an error if these rules are violated; such as when `hk` has been declared as an integer, or when the character string contains more than 70 characters.

3.4 Testing

Testing was carried out continuously during the design and implementation of the input format and the translator. The NJOY test problems[1] [1] was used to test the functionality of both the input format and the translator.

The NJOY test problems was manually translated into equivalent NJOY jobs in the new input format, which were run through the translator. The resulting output was compared with the expected output, to verify that the translator was working appropriately.

The Python unit testing framework [9] was utilized to set up the testing environment.

[1]The NJOY Test Problems are test runs which are used to test the functionality of the NJOY software system. See `http://t2.lanl.gov/codes/njoy99/`

4 Implementation

4.1 NJOY Input Format (`NIF`)

The new input format, NJOY Input Format (`NIF`), is basically the original NJOY input instructions which have been annotated with a syntax to make it easier to read and express. NIF has been designed to appear more like a high-level programming language.

4.1.1 Grammar Definition

The proposed NJOY Input Format (`NIF`) is illustrated as a context-free grammar definition in algorithm 3 on the next page. The structure of the grammar is simple. Just like in reference [4], a NIF program is an ordered sequence of modules. Each module is composed by an ordered sequence of cards. A card is an ordered sequence of value definitions.

In NIF, the start symbol is `program`. The capitalized terminals, such as `MODULE` and `CARD`, are token classes specified by the lexer. Special symbols are denoted within double quotes. `empty` denotes the empty string.

An assignment denotes that a left hand side is assigned to hold the values of a right hand side. A left hand side is an ordered list of elements, where the elements can be an array or identifier. A right hand side is an ordered list of elements, where the elements can be a float, integer, null or a string. As such, a value definition is an array or identifier that has been declared to hold the value of either a floating-point number, natural number, empty string or a character string.

As indicated by the grammar, NIF supports multiple assignment. That is, multiple identifiers can be assigned in the same expression. For example, the expression

```
material, temp = 9237, 300.0;
```

denotes that the identifier `material` holds the integer 9237, and the identifier `temp` holds the float 300.0. The syntax analysis in the parser enforces that the number of elements on both sides of an assignment are the same.

Algorithm 3 NJOY Input Format (NIF) Grammar Definition

```
program ::= module_list

module_list ::= module module_list
              | empty
module      ::= MODULE "{" card_list "}"

card_list ::= card card_list
            | empty
card      ::= CARD "{" stmt_list "}"

stmt_list  ::= statement stmt_list
             | empty
statement  ::= expression ";"
expression ::= assignment

assignment ::= l_value_list "=" r_value_list

l_value_list ::= l_value
               | l_value "," l_value_list
r_value_list ::= r_value
               | r_value "," r_value_list

l_value ::= array
          | ident
array   ::= IDENTIFIER "[" INTEGER "]"
ident   ::= IDENTIFIER

r_value ::= FLOAT
          | INTEGER
          | NULL
          | STRING
```

4.2 NJOY Input Format Translator (`nifty`)

4.2.1 Structure of the Translator

The translator, NJOY Input Format Translator (`nifty`), was constructed as a set of modules where each module implements a specific phase in the translation process. Five phases have been implemented as part of the translation process and are shown in figure 1.

The first phase is the lexical analysis which is implemented by the lexer module. The second phase, syntax analysis, is implemented by the parser module.

The third phase, implemented by the organizer module, is a special phase where the order of the statements in a card are analyzed and possible rearranged.

The fourth phase is the semantic analysis which is implemented by the module named analyzer. The fifth, and final, phase of the translator is the emitter module which implements a NJOY input instructions generator.

A basic user manual for the translator is available in appendix A on page 26.

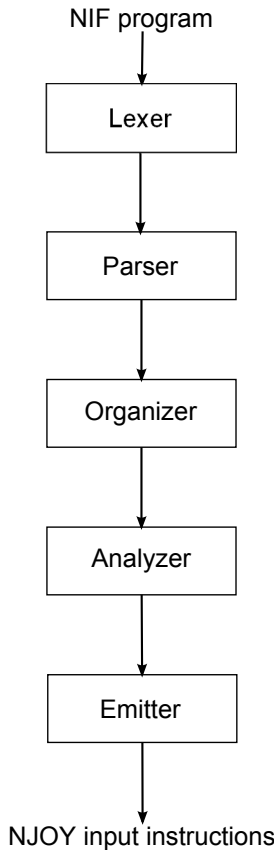

Figure 1: Translation process in `nifty`

4.2.2 Reserved Keywords

An important design choice is that the translator will enforce the use of reserved keywords to specify NIF programs. It will not only consider card and module names as reserved keywords, but also identifier names. As such, it is not possible to use an identifier name until it has been defined as an identifier in the translator. Similarly, it is not possible to use a card or module name which has not been defined in the translator. This restricts the expressiveness of the input format, but allows detailed analysis of the semantics in the organizer and analyzer modules. As a consequence, it also forces the user to write consistent and readable input files – which has been the objective of this work.

4.2.3 The Modules

Lexer The lexer is responsible for recognizing character patterns and generating the appropriate NIF tokens. As input, the lexer expects a NIF program and will generate a token stream as its output unless the lexer detects a lexical error. If a lexical error is detected, an error message will be reported and the translation process will stop at this phase. The lexer will only recognize card and module names which are specified in reference [4], thus enforcing the use of a specific set of cards and modules as mentioned previously. The lexer also recognizes comments in the input program. The comments will be discarded during the lexical analysis and thus won't be passed on to the next phase in the translation process.

Parser The parser is responsible for enforcing the structure of the NIF grammar and constructing the syntax tree. As input, the parser expects a stream of tokens generated by the lexer. The parser will produce a syntax tree as its output, which represents the structure of the NIF program. If the parser detects a syntax error, an error message will be reported and the translation process will stop at this phase.

Organizer The organizer analyzes the syntax tree produced by the parser. Its purpose is to rearrange the statements in a card such that they appear in the expected, working order. As such, it should be possible to write a NIF program without having to list the statements in a card in the expected order as indicated by reference [4].

The NJOY modules and the cards within the modules still needs to be given in the correct order though. This is due to the fact that the number of possible NJOY jobs is infinite (all may not be functional in the NJOY software system, though). An infinite number of NJOY jobs can simply be created by just appending another module specification to an existing NJOY job in order to create a new one. Simply stated, the translator can not guess the intention of the job due to the number of possible combinations the modules may be listed in. Hence, the modules must be provided in the expected order by the user. Cards are not arrangeable either, since they also are prone to be repetitive. It is not possible to determine which card should go first from a set of cards (with the same name) which e.g. only contains a descriptive title. The cards must also be provided in the expected order by the user.

Each NJOY module requires its own organizer implementation since each module has its specific set of rules as described in reference [4]. Since the identifier names are hardwired in the translator, the organizer is able to do a detailed analysis of the syntax tree and easily detect if a specific identifier has been defined out of order.

If any statements have been provided out of order in a card, and the organizer is able to arrange the statements, a new syntax tree is returned where the statements have been ordered in the expected sequence. If the organizer somehow fails to organize the syntax tree, it will return the original syntax tree

as produced by the parser and pass it on to the next phase in the translation process.

Analyzer The analyzer expects a syntax tree as its input. Like in the organizer phase, the NJOY modules needs to be analyzed separately since each module has its specific set of rules. As such, each module also requires its own analyzer implementation.

The analyzer basically visits every node in the order they appear in the syntax tree and checks if it is the expected one. The analysis can be made very detailed since the translator can, to some extent, predict the next card or identifier due to the ordered nature described in reference [4]. Since the cards and the identifiers have reserved names, the analyzer is able to easily determine whether a card or an identifier is the expected one. Using reserved names also makes type checking easy, since a reserved identifier in a specific card may be associated with a specific type, range, size, length, et cetera.

The analyzer does not alter the syntax tree, it just analyzes it. The input syntax tree will be the output of the analyzer if the syntax tree is semantically correct according to the translator. If the analyzer detects a semantic error in the syntax tree, an error message will be reported and the translation process will stop at this phase.

Emitter The emitter expects a syntax tree as its input and it is responsible for generating NJOY input instructions from the syntax tree. The emitter simply flattens the tree structure and formats the instructions to their corresponding counterparts in the NJOY input instructions format. The emitter returns a string with the resulting NJOY input instructions. Each card in the resulting output has been annotated with a descriptive comment, indicating which card it is, to make it easier to find errors.

4.3 Translation Verification

As previously described in section 3.4 on page 7, the NJOY test problems [1] were manually translated into NIF programs expressing the equivalent NJOY jobs. The resulting NIF programs were used for verifying that the implementation was working appropriately by setting up a test suite using the Python unit testing framework [9]. Each test problem was set up to be run through each individual phase in the translation process, and each run were expected to be successful since the NIF programs should be an equivalent and functional version of the original test problems.

The resulting output, as produced by the emitter, was compared with the expected output. That is, each NIF version of the test problems were compared with its corresponding original NJOY test problem.

Note that modified versions of the original NJOY test problems had to be used as the expected output when comparing the resulting output from the translator. The comments that the emitter appends to every card had to be appended to the expected output such that the comparison could be performed.

4.4 Translation Efficiency

A simple Python script was written to check the resulting translation efficiency of the translator. Two different notions of timing were used in the script, namely *process time* and *wall time*.

Process time is the time that the entire task spent executing on the processor, measured by `time.clock()` which should be used for timing algorithms [9]. Wall time is the time that elapsed from when the task was started to when the task finished, measured by checking the difference in time using `time.time()` [9]. The main difference between process time and wall time is that wall time is the time it takes until the system delivers the computed result, whereas process time is the time that it took to compute the result.

5 Results

5.1 NJOY Input Format (NIF)

The result of the proposed grammar described in section 4.1 on page 8 is best illustrated with examples. Algorithm 4 illustrates NJOY input instructions (slightly modified to make it shorter for illustrational purposes) from NJOY Test Problem 2 [1]. In algorithm 5 on the following page, lines 1 through 9 from algorithm 4 are expressed in NIF.

Algorithm 4 Modified subset of NJOY Test Problem 2

```
1  moder
2  20 -21/
3  reconr
4  -21 -22/
5  'pendf tape for pu-238 from endf/b-iv tape 404'/
6  1050 1/
7  0.005/
8  '94-pu-238 from endf/b tape t404'/
9  0/
10 broadr
11 -21 -22 -23/
12 1050 3 0 1/
13 0.005/
14 300.0 900.0 2100.0/
15 0/
16 stop
```

Algorithm 5 NIF version of Algorithm 4 on the preceding page, lines 1 through 9

```
1  moder {
2      card_1 {
3          pendf_input = 20;
4          pendf_output = -21;
5      }
6  }
7
8  reconr {
9      card_1 {
10         nendf = -21;
11         npend = -22;
12     }
13
14     card_2 {
15         tlabel =  "pendf tape for pu-238 from endf/b-
               iv tape 404";
16     }
17
18     card_3 {
19         mat = 1050;
20         ncards = 1;
21     }
22
23     card_4 {
24         err = 0.005;
25     }
26
27     card_5 {
28         cards = "94-pu-238 from endf/b tape t404";
29     }
30     /* Card 6 not defined since 'ngrid' defaults to 0
          in first card 3. */
31     card_3 { mat = 0; } // Terminate reconr.
32 }
```

Descriptive names for the identifiers on line 3 and 4 have been specified in the translator. The other identifier names has been chosen to reflect the documentation in reference [4], but the identifier names are interchangeable in the translator (the identifier names are hardwired in the translator). Line 30 and 31 shows how comments are expressed in NIF. Line 30 illustrates the structure of multiline comments while line 31 illustrates the structure of single line comments.

Algorithm 6 is a NIF version of the lines 10 through 16 from algorithm 4 on page 14. It shows how arrays are expressed in NIF (lines 24 through 26). The `stop` instruction on line 16 in algorithm 4 on page 14 does not have to be specified in NIF, the translator will automatically append it in the translation process.

When combined, algorithm 5 on the preceding page and algorithm 6 forms the complete NJOY job as listed in algorithm 4 on page 14.

Algorithm 6 NIF version of Algorithm 4 on page 14, lines 10 through 16

```
1  broadr {
2      card_1
3      {
4          nendf = -21;
5          nin = -22;
6          nout = -23;
7      }
8
9      card_2
10     {
11         mat1 = 1050;
12         ntemp2 = 3;
13         istart = 0;
14         istrap = 1;
15     }
16
17     card_3
18     {
19         errthn = 0.005;
20     }
21
22     card_4
23     {
24         temp2[0] = 300.0;
25         temp2[1] = 900.0;
26         temp2[2] = 2100.0;
27     }
28
29     /* Terminate execution of broadr with mat1 = 0 as
          usual. */
30     card_5
31     {
32         mat1 = 0;
33     }
34 }
```

5.2 NJOY Input Format Translator (`nifty`)

Table 2 shows the implementation status for the NJOY modules. Each column entry indicates the completeness of a translator phase for a given NJOY module.

NJOY Module	Lexer	Parser	Organizer	Analyzer	Emitter
acer	100%		100%	90%	100%
broadr			100%	90%	
ccccr			0%		
covr			100%	90%	
dtfr			0%		
errorr			70%	20%	
gaminr			100%	90%	
gaspr			100%	99%	
groupr			100%	90%	
heatr			100%	90%	
leapr			0%		
matxsr					
mixr					
moder			100%	95%	
plotr			100%	90%	
powr			0%		
purr			100%	90%	
reconr			100%	90%	
resxsr			0%		
thermr			100%	90%	
unresr			100%	95%	
viewr			100%	10%	
wimsr			0%		

Table 2: Implementation status for the NJOY modules

The completeness of the implementation has been rated in a grading scale with percentage. The grades has been set with respect to whether the functionality of the phases presented in section 4.2.3 on page 11 (also see section 3.3 on page 6) has been fulfilled or not. 100% indicates that the functionality has been finished. 0% indicates that the implementation of the functionality has not been started. The other percentages are rough approximations of how much functionality that has been implemented.

5.3 Translation Verification

The result of the translation verification is summarized in table 3. All test problems listed in Appendix B on page 29 passed all the phases in the translation process. That is, the test problems were successfully translated[2]; no lexical, syntax, nor semantic errors were found. No differences between the expected output and the resulting output were detected for the test problems.

Test Problem	Translator Phases	Output
tp01	Passed	Expected
tp02		
tp03		
tp04		
tp05		
tp06		
tp07		
tp08		
tp10		
tp11		
tp12		
tp13		
tp14		
tp17		

Table 3: Translation verification results for the test problems

[2]Note that the organizer's ability to arrange statements in the correct order has not been tested for the test problems, since the instructions in the test problems have been provided in the expected order.

5.4 Translation Efficiency

The efficiency of the translator was tested by running the entire translation process for each test problem, listed in Appendix B on page 29, 10 000 times. Table 4 shows the resulting runtimes, both process time and wall time, in seconds. The resulting runtimes denotes the aggregate of 10 000 repeated runs for a given test problem.

Test Problem	**Process Time**	**Wall Time**
`tp01`	2.87	345.22
`tp02`	2.87	374.29
`tp03`	2.81	292.49
`tp04`	2.79	281.89
`tp05`	3.10	255.14
`tp06`	3.05	346.87
`tp07`	3.07	278.47
`tp08`	2.75	294.17
`tp10`	3.05	288.98
`tp11`	2.84	373.76
`tp12`	2.72	301.82
`tp13`	3.19	280.76
`tp14`	3.25	251.40
`tp17`	2.95	350.35
Average Time:	2.95	308.26

Table 4: Aggregated runtimes (in seconds) for 10 000 runs

The average wall time for a single run for the test problems is $308.26/10000 \approx 0.031$ seconds.

The repeated runs were conducted on a multi-user system equipped with three Dual Core AMD Opteron Processor 280 at 2.4GHz each, and a total of 3.6GB RAM. The system was running Linux 2.6.18 and Python 2.4.3.

The Python library functions `time.clock()` and `time.time()` [9] were used to measure the process time and wall time, respectively.

6 Discussion

6.1 NJOY Input Format (`NIF`)

As indicated by the examples listed in section 5.1 on page 14, a typical NIF program is vertically long compared to the compact notation of the NJOY input instructions. NIF programs can of course be specified in a compact form as well, e.g. on a single line, but this is not the intended usage of NIF. The purpose of NIF is to make NJOY jobs readable. The readability would be limited if the jobs were expressed on a single line.

The proposed grammar does not differ much from the original NJOY input instructions since it basically is an annotated version of them. The NIF grammar could have been expanded to include more complex programming idioms, such as an `if` expression to allow flow control in a NIF program. Although, the structure of NIF was designed to be simple and to closely resemble the original input instructions such that a user does not need to learn a completely new programming language to specify NJOY jobs. Another intention of this design choice is that the NJOY input instructions documented in reference [4] can be used to specify NJOY jobs in NIF.

6.2 NJOY Input Format Translator (`nifty`)

The translator is able to translate NIF programs into functional NJOY input instructions, but an organizer and analyzer has not been provided for all modules in the NJOY software system due to time constraints of this thesis. As such, the important semantic analysis of the translator is incomplete. However, much of the needed functionality and structure is provided by the existing implementation such that both the organizer and the analyzer should be easy to complete.

The implementation of the analyzer module has been the most time consuming task when designing the translator. It requires detailed analysis of what kind of input the NJOY modules expect and how they operate on it. The documentation in reference [4] was the main resource used while implementing the semantic analysis in the analyzer. It was evident that this was not a sufficient resource for the task at hand. It does not clearly indicate the expected type for all identifiers, nor the expected integer ranges or length of the character strings. In some cases, it has also been hard to deduce which cards that must be supplied by just reading the documentation in reference [4]. To fully check the semantics of a NIF program, the source code for the NJOY software system must be studied in greater detail. The ENDF formats must also be studied in greater detail in order to understand the semantics and what kind of values that the NJOY modules accept.

6.3 Translation Verification

The testing that was conducted within this work is not rigorous enough due to time constraints of this thesis. NJOY is a large and complex program[3] with many possible combinations of input within each NJOY module and card. The NJOY test problems [1] which were used to test the translation functionality is a very small set of possible NJOY jobs. Hence, there is a lot of scenarios within each NJOY module that has not been tested.

6.4 Translation Efficiency

The efficiency testing of the translator as described in section 5.4 on page 19 was conducted in a simple fashion. The resulting process runtimes revealed that the performance of the translator implementation is not a huge bottleneck, compared to the much greater wall times. The performance appeared to be good enough to fit the purpose of the translator. Therefore, more elaborate testing of the efficiency was not conducted.

[3] The source files for the NJOY software system consists of more than 100 000 lines.

7 Conclusions

In this thesis, a new input format, NJOY Input Format (`NIF`), has been designed. A translator which is able to translate NIF into NJOY input instructions has been implemented.

It is possible to specify basic NJOY jobs in NIF. The resulting NIF programs can be translated into NJOY input instructions, which can be run by the NJOY software system. Production use is although not advisable, since it has been challenging to conduct rigorous and complete testing.

It has also been evident that analyzing the NJOY input instructions is not enough to design a new input format for the NJOY software system. Analyzing the ENDF libraries and the source code for the NJOY software system is required in order to build a translator which can conduct a complete semantic analysis for an NJOY job.

8 Future Work

Future work includes completing the semantic analysis and the organizer feature for all modules in the NJOY software system. The NJOY Input Format and the translator also needs to be systematically evaluated and verified by a complete software quality assurance process as described in reference [15]. The project may also be expanded to include default scenarios, which uses normal mode of operation per default, such that the user does not have to specify exhaustive NJOY jobs just to convert a library into another.

A spin-off project, that is related to developing a user friendly and readable input format, is to construct a graphical user interface editor which can display and produce NJOY input instructions in a user friendly fashion.

References

[1] R. E. MacFarlane, "NJOY99 – code system for producing pointwise and multigroup neutron and photon cross-sections from ENDF/B data", Los Alamos Nat. Laboratory, Los Alamos, NM, Rep. RSIC PSR-480, 2000.

[2] M. B. Chadwick *et al.*, "ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology," *Nuclear Data Sheets*, vol. 107, no. 12, pp. 2931-3060, Dec. 2006.

[3] C. Gustavsson *et al.*, "Massive Computation Methodology for Reactor Operation (MACRO)," in *European Nuclear Conference*, 2010 © European Nuclear Society. ISBN: 978-92-95064-09-6

[4] A. C. Kahler and R. E. MacFarlane. (2010, Mar. 31). *User Input for NJOY99, updated through version 364* [Online]. Available: `http://t2.lanl.gov/codes/njoy99/Userinp.364`

[5] A. V. Aho *et al.*, *Compilers: Principles, Techniques, & Tools*, Second Edition. Boston: Pearson Educ., 2007.

[6] A. V. Aho *et al.*, "Syntax Analysis" in *Compilers: Principles, Techniques, & Tools*, Second Edition. Boston: Pearson Educ., 2007, ch. 4, sec. 4.2, pp. 197-206.

[7] A. V. Aho *et al.*, "Lexical Analysis" in *Compilers: Principles, Techniques, & Tools*, Second Edition. Boston: Pearson Educ., 2007, ch. 3, sec. 3.5, pp. 140-146.

[8] A. V. Aho *et al.*, "Syntax Analysis" in *Compilers: Principles, Techniques, & Tools*, Second Edition. Boston: Pearson Educ., 2007, ch. 4, sec. 4.9, pp. 287-297.

[9] F. L. Drake, Jr., *et al.* (2011, Apr. 16) *Python v2.7.1 documentation* [Online]. Available: `http://docs.python.org/`

[10] A. V. Aho *et al.*, "Lexical Analysis" in *Compilers: Principles, Techniques, & Tools*, Second Edition. Boston: Pearson Educ., 2007, ch. 3, sec. 3.1, pp. 109-114.

[11] D. M. Beazley. (2011, Apr. 16). *PLY (Python Lex-Yacc)* [Online]. Available: `http://www.dabeaz.com/ply/ply.html`

[12] A. V. Aho *et al.*, "Lexical Analysis" in *Compilers: Principles, Techniques, & Tools*, Second Edition. Boston: Pearson Educ., 2007, ch. 3, sec. 3.3, pp. 116-124.

[13] A. V. Aho *et al.*, "Syntax Analysis" in *Compilers: Principles, Techniques, & Tools*, Second Edition. Boston: Pearson Educ., 2007, ch. 4, sec. 4.1, pp. 192-196.

[14] A. V. Aho *et al.*, "Introduction" in *Compilers: Principles, Techniques, & Tools*, Second Edition. Boston: Pearson Educ., 2007, ch. 1, sec. 1.2, pp. 8-9.

[15] C. Kaner *et al.*, *Testing Computer Software*, Second Edition. New York: John Wiley and Sons, Inc., 1999.

A User Manual

A.1 Structure of `nifty`

The `nifty` directory structure is organized as shown in figure A.1.

```
nifty/
    bin/
        analyzer
        bench
        emitter
        lexer
        nifty
        organizer
        parser
        test
    data/
        ...
        test_problems/
    nifty/
        analyzer/
        emitter/
        environment/
        lexer/
        organizer/
        parser/
        settings/
        tests/
    [ply/]
```

Figure A.1: Directory Structure of `nifty`

The `nifty/bin/` directory includes all executable Python scripts which are used for running and testing the translator. The `nifty` executable in the `nifty/bin/` directory runs the complete translation process on an input NIF program. The `test` executable runs the test suite. The `bench` executable is a script used for testing the efficiency of the translator. The other executable scripts runs their corresponding named phase in the translation process (and all the successive phases that they depend on).

The test problems are located in the `nifty/data/test_problems/` directory. The `nifty/nifty/` directory contains the source code for the translator. The optional directory `ply/` indicates where PLY can be placed such that the translator is able to locate it.

A.2 Installation

Python version 2.2 or greater is required to use `nifty`. Python version 2.4.3 and 2.6.1 has been tested with `nifty` and are known to work. `nifty` itself does not require any special installation methods, although PLY [11] is required to run the translator. It is sufficient to download PLY and put the `ply/` directory from PLY in the `nifty/` top directory as indicated by figure A.1 on the previous page. (Note the non-restrictive license of PLY generously provided by its author.)

A.3 Running the Translator

The translator has been implemented as a command-line based interface for a Unix-like environment. To run the entire translation process, the `nifty` executable in the `nifty/bin/` directory should be used. Issuing the command

```
bin/nifty -h
```

in the `nifty/` top directory, will print the usage message shown in figure A.2.

```
usage: nifty [options] [input_file] [output_file]
options:
    -h, --help  show this help message and exit
    -a          don't analyze the input
    -o          don't organize the input
```

Figure A.2: `bin/nifty` usage

The `options` flag(s) are optional. The `input_file` and `output_file` are also optional. If no input file is given, standard input (`stdin`) will be used as the input source. If no output file is given, the result will be redirected to standard output (`stdout`).

As an example, the command

```
bin/nifty input.nif output
```

will simply run the translator on a file named `input.nif` and output the resulting NJOY input instructions on a file named `output`. The analyzer and organizer phase can be skipped by giving the `-a` and `-o` flag

`bin/nifty -a input.nif output`, to skip the analyzer phase

`bin/nifty -o input.nif output`, to skip the organizer phase

To skip both the organizer and analyzer phase, run `nifty` with both flags specified

```
bin/nifty -ao input.nif output
```

A.4 Settings

The `nifty/nifty/settings/` directory as shown in figure Figure A.1 on page 26 contains module specific settings. Each module has its own settings file. The analyzer and organizer phase utilizes the settings files when processing a syntax tree. The settings describes the expected identifiers and their expected order within the cards in a module.

B Test Problems

In this section, the test problems that were used for testing the functionality of the translator is listed. Both the NIF versions and the expected NJOY input instructions are provided. The test problems listed in this section are also available in the `nifty/data/test_problems/` directory, as described in section A.1 on page 26.

B.1 Test Problem 01 (`tp01`)

NIF Version of Test Problem 01

```
1  moder
2  {
3      card_1
4      {
5          nin = 20;
6          nout = -21;
7      }
8  }
9
10 reconr
11 {
12     card_1
13     {
14         nendf = -21;
15         npend = -22;
16     }
17
18     card_2
19     {
20         tlabel =  "pendf tape for c-nat from endf/b tape 511";
21     }
22
23     card_3
24     {
25         mat = 1306;
26         ncards = 3;
27     }
28
29     card_4
30     {
31         err = 0.005; // Use C-style floats.
32     }
33
34     card_5
35     {
36         cards = "6-c-nat from tape 511";
37     }
38
39     card_5
40     {
41         cards = "processed by the njoy nuclear data processing system";
42     }
43
44     card_5
45     {
46         cards = "see original endf/b-v tape for details of evaluation";
47     }
48
49     /* Card 6 skipped since ngrid defaults to 0 in first card 3 */
50
51     card_3
52     {
```

```
53          mat = 0;
54      }
55  }
56
57  broadr
58  {
59      card_1
60      {
61          nendf = -21;
62          nin = -22;
63          nout = -23;
64      }
65
66      card_2
67      {
68          mat1 = 1306;
69          ntemp2 = 1;
70      }
71
72      card_3
73      {
74          errthn = 0.005; // Use C-style floats.
75      }
76
77      card_4
78      {
79          temp2[0] = 300.0; // Use C-style floats.
80      }
81
82      card_5
83      {
84          mat1 = 0;
85      }
86  }
87
88  heatr
89  {
90      card_1
91      {
92          nendf = -21;
93          nin = -23;
94          nout = -22;
95      }
96
97      card_2
98      {
99          matd = 1306;
100         npk = 1;
101     }
102
103     card_3
104     {
105         mtk[0] = 444; // Note that mtk has to be defined as an array.
106     }
107
108     /* Card 4, 5, and 5a are skipped since nqa defaults to 0 in card 2. */
109 }
110
111 thermr
112 {
113     card_1
114     {
115         nendf = 0;
116         nin = -22;
117         nout = -24;
118     }
119
120     card_2
```

```
    {
        matde = 0;
        matdp = 1306;
        nbin = 8;
        ntemp = 1;
        iinc = 1;
        icoh = 0;
        natom = 1;
        mtref = 221;
        iprint = 0;
    }

    card_3
    {
        tempr[0] = 300.0; // Use C-style floats.
    }

    card_4
    {
        tol = 0.05; // Use C-style floats.
        emax = 1.2;
    }
}

thermr
{
    card_1
    {
        nendf = 26;
        nin = -24;
        nout = -23;
    }

    card_2
    {
        matde = 1065;
        matdp = 1306;
        nbin = 8;
        ntemp = 1;
        iinc = 4;
        icoh = 1;
        natom = 1;
        mtref = 229;
        iprint = 0;
    }

    card_3
    {
        tempr[0] = 300.0; // Use C-style floats.
    }

    card_4
    {
        tol = 0.05; // Use C-style floats.
        emax = 1.2;
    }
}

groupr
{
    card_1
    {
        nendf = -21;
        npend = -23;
        ngout1 = 0;
        ngout2 = -24;
    }

```

```
    card_2
    {
        matb = 1306;
        ign = 3;
        igg = 3;
        iwt = 3;
        lord = 3;
        ntemp = 1;
        nsigz = 1;
        iprint = 1;
    }

    card_3
    {
        title = "carbon in graphite";
    }

    card_4
    {
        temp[0] = 300;
    }

    card_5
    {
        sigz[0] = 1.0e10; // No trailing dots. Use C-style floats.
    }

    card_9
    {
        mfd = 3;
        mtd = 1;
        mtname = "total";
    }

    card_9
    {
        mfd = 3;
        mtd = 2;
        mtname = "elastic";
    }

    card_9
    {
        mfd = 3;
        mtd = 4;
        mtname = "inelastic";
    }

    card_9
    {
        mfd = 3;
        mtd = 51;
        mtname = "discrete inelastic";
    }

    card_9
    {
        mfd = 3;
        mtd = -68;
        mtname = "continued";
    }

    card_9
    {
        mfd = 3;
        mtd = 91;
        mtname = "continuum inelastic";
    }
```

```
257
258       card_9
259       {
260           mfd = 3;
261           mtd = 102;
262           mtname = "n,g";
263       }
264
265       card_9
266       {
267           mfd = 3;
268           mtd = 103;
269           mtname = "(n,p)";
270       }
271
272       card_9
273       {
274           mfd = 3;
275           mtd = 104;
276           mtname = "(n,d)";
277       }
278
279       card_9
280       {
281           mfd = 3;
282           mtd = 107;
283           mtname = "(n,a)";
284       }
285
286       card_9
287       {
288           mfd = 3;
289           mtd = 221;
290           mtname = "free thermal scattering";
291       }
292
293       card_9
294       {
295           mfd = 3;
296           mtd = 229;
297           mtname = "graphite inelastic thermal scattering";
298       }
299
300       card_9
301       {
302           mfd = 3;
303           mtd = 230;
304           mtname = "graphite elastic thermal scattering";
305       }
306
307       card_9
308       {
309           mfd = 3;
310           mtd = 251;
311           mtname = "mubar";
312       }
313
314       card_9
315       {
316           mfd = 3;
317           mtd = 252;
318           mtname = "xi";
319       }
320
321       card_9
322       {
323           mfd = 3;
324           mtd = 253;
```

```
325         mtname = "gamma";
326     }
327
328     card_9
329     {
330         mfd = 3;
331         mtd = 301;
332         mtname = "total heat production";
333     }
334
335     card_9
336     {
337         mfd = 3;
338         mtd = 444;
339         mtname = "total damage energy production";
340     }
341
342     card_9
343     {
344         mfd = 6;
345         mtd = 2;
346         mtname = "elastic";
347     }
348
349     card_9
350     {
351         mfd = 6;
352         mtd = 51;
353         mtname = "discrete inelastic";
354     }
355
356     card_9
357     {
358         mfd = 6;
359         mtd = -68;
360         mtname = "continued";
361     }
362
363     card_9
364     {
365         mfd = 6;
366         mtd = 91;
367         mtname = "continuum inelastic";
368     }
369
370     card_9
371     {
372         mfd = 6;
373         mtd = 221;
374         mtname = "free thermal scattering";
375     }
376
377     card_9
378     {
379         mfd = 6;
380         mtd = 229;
381         mtname = "graphite inelastic thermal scattering";
382     }
383
384     card_9
385     {
386         mfd = 6;
387         mtd = 230;
388         mtname = "graphite elastic thermal scattering";
389     }
390
391     card_9
392     {
```

```
393           mfd = 17;
394           mtd = 51;
395           mtname = "inelastic gamma production";
396       }
397
398       card_9
399       {
400           mfd = 16;
401           mtd = 102;
402           mtname = "capture gamma production";
403       }
404
405       card_9
406       {
407           mfd = 0;
408       }
409
410       card_10
411       {
412           matd = 0;
413       }
414   }
415
416   moder
417   {
418       card_1
419       {
420           nin = -23;
421           nout = 25;
422       }
423   }
```

Expected NJOY Input Instructions for Test Problem 01

```
1   moder
2   20 -21/ ### card_1
3   reconr
4   -21 -22/ ### card_1
5   'pendf tape for c-nat from endf/b tape 511'/ ### card_2
6   1306 3/ ### card_3
7   0.005/ ### card_4
8   '6-c-nat from tape 511'/ ### card_5
9   'processed by the njoy nuclear data processing system'/ ### card_5
10  'see original endf/b-v tape for details of evaluation'/ ### card_5
11  0/ ### card_3
12  broadr
13  -21 -22 -23/ ### card_1
14  1306 1/ ### card_2
15  0.005/ ### card_3
16  300.0/ ### card_4
17  0/ ### card_5
18  heatr
19  -21 -23 -22/ ### card_1
20  1306 1/ ### card_2
21  444/ ### card_3
22  thermr
23  0 -22 -24/ ### card_1
24  0 1306 8 1 1 0 1 221 0/ ### card_2
25  300.0/ ### card_3
26  0.05 1.2/ ### card_4
27  thermr
28  26 -24 -23/ ### card_1
29  1065 1306 8 1 4 1 1 229 0/ ### card_2
30  300.0/ ### card_3
31  0.05 1.2/ ### card_4
32  groupr
33  -21 -23 0 -24/ ### card_1
```

```
1306 3 3 3 3 1 1 1/ ### card_2
'carbon in graphite'/ ### card_3
300/ ### card_4
1.0e10/ ### card_5
3 1 'total'/ ### card_9
3 2 'elastic'/ ### card_9
3 4 'inelastic'/ ### card_9
3 51 'discrete inelastic'/ ### card_9
3 -68 'continued'/ ### card_9
3 91 'continuum inelastic'/ ### card_9
3 102 'n,g'/ ### card_9
3 103 '(n,p)'/ ### card_9
3 104 '(n,d)'/ ### card_9
3 107 '(n,a)'/ ### card_9
3 221 'free thermal scattering'/ ### card_9
3 229 'graphite inelastic thermal scattering'/ ### card_9
3 230 'graphite elastic thermal scattering'/ ### card_9
3 251 'mubar'/ ### card_9
3 252 'xi'/ ### card_9
3 253 'gamma'/ ### card_9
3 301 'total heat production'/ ### card_9
3 444 'total damage energy production'/ ### card_9
6 2 'elastic'/ ### card_9
6 51 'discrete inelastic'/ ### card_9
6 -68 'continued'/ ### card_9
6 91 'continuum inelastic'/ ### card_9
6 221 'free thermal scattering'/ ### card_9
6 229 'graphite inelastic thermal scattering'/ ### card_9
6 230 'graphite elastic thermal scattering'/ ### card_9
17 51 'inelastic gamma production'/ ### card_9
16 102 'capture gamma production'/ ### card_9
0/ ### card_9
0/ ### card_10
moder
-23 25/ ### card_1
stop
```

B.2 Test Problem 02 (tp02)

NIF Version of Test Problem 02

```
moder
{
    card_1
    {
        nin = 20;
        nout = -21;
    }
}

reconr
{
    card_1
    {
        nendf = -21;
        npend = -22;
    }

    card_2
    {
        tlabel =  "pendf tape for pu-238 from endf/b-iv tape 404";
    }

    card_3
    {
        mat = 1050;
        ncards = 3;
    }

    card_4
    {
        err = 0.005; // Use C-style floats instead of ".005".
    }

    card_5
    {
        cards = "94-pu-238 from endf/b tape t404";
    }

    card_5
    {
        cards = "processed by the njoy nuclear data processing system";
    }

    card_5
    {
        cards = "see original endf/b-iv tape for details of evaluation";
    }

    /* Card 6 skipped since ngrid defaults to 0 in first card 3. */

    card_3
    {
        mat = 0;
    }
}

broadr
{
    card_1
    {
        nendf = -21;
        nin = -22;
        nout = -23;
    }
```

```
65
66      card_2
67      {
68          mat1 = 1050;
69          ntemp2 = 3;
70          istart = 0;
71          istrap = 1;
72          temp1 = 0;
73      }
74
75      card_3
76      {
77          errthn = 0.005; // Use C-style floats instead of ".005".
78      }
79
80      card_4
81      {
82          /* In this example, Each temperature is declared as an element in an
83             array.
84             ntemp2 in card_2 denotes the number of expected temperatures.
85           */
86          temp2[0] = 300.0;
87          temp2[1] = 900.0;
88          temp2[2] = 2100.0;
89      }
90
91      card_5
92      {
93          mat1 = 0;
94      }
95  }
96
97  moder
98  {
99      card_1
100     {
101         nin = -23;
102         nout = 33;
103     }
104 }
105
106 unresr
107 {
108     card_1
109     {
110         nendf = -21;
111         nin = -23;
112         nout = -24;
113     }
114
115     card_2
116     {
117         matd = 1050;
118         ntemp = 3;
119         nsigz = 7;
120         iprint = 1;
121     }
122
123     card_3
124     {
125         temp[0] = 300;
126         temp[1] = 900;
127         temp[2] = 2100;
128     }
129
130     card_4
131     {
132         sigz[0] = 1.0e10;
```

```
133           sigz[1] = 1.0e5;
134           sigz[2] = 1.0e4;
135           sigz[3] = 1000.0;
136           sigz[4] = 100.0;
137           sigz[5] = 10.0;
138           sigz[6] = 1;
139       }
140
141       card_2
142       {
143           matd = 0;
144       }
145   }
146
147   groupr
148   {
149       card_1
150       {
151           nendf = -21;
152           npend = -24;
153           ngout1 = 0;
154           ngout2 = -25;
155       }
156
157       card_2
158       {
159           matb = 1050;
160           ign = 5;
161           igg = 0;
162           iwt = 4;
163           lord = 3;
164           ntemp = 3;
165           nsigz = 7;
166           iprint = 1;
167       }
168
169       card_3
170       {
171           title = "94-pu-238";
172       }
173
174       card_4
175       {
176           /* ntemp in card_2 denotes the number of expected temperatures. */
177           temp[0] = 300.0;
178           temp[1] = 900.0;
179           temp[2] = 2100.0;
180       }
181
182       card_5
183       {
184           /* nsigz in card_2 denotes the number of expected sigma zeroes. */
185           sigz[0] = 1.0e10;
186           sigz[1] = 1.0e5;
187           sigz[2] = 1.0e4;
188           sigz[3] = 1000.0;
189           sigz[4] = 100.0;
190           sigz[5] = 10.0;
191           sigz[6] = 1;
192       }
193
194       card_8c
195       {
196           eb = 0.1;
197           tb = 0.025;
198           ec = 0.8208e06;
199           tc = 1.4e06;
200       }
```

```
201
202     /* Reactions for temperature 300.0. */
203     card_9
204     {
205         mfd = 3;
206         mtd = 1;
207         mtname = "total";
208     }
209
210     card_9
211     {
212         mfd = 3;
213         mtd = 2;
214         mtname = "elastic";
215     }
216
217     card_9
218     {
219         mfd = 3;
220         mtd = 16;
221         mtname = "n2n";
222     }
223
224     card_9
225     {
226         mfd = 3;
227         mtd = 17;
228         mtname = "n3n";
229     }
230
231     card_9
232     {
233         mfd = 3;
234         mtd = 18;
235         mtname = "fission";
236     }
237
238     card_9
239     {
240         mfd = 3;
241         mtd = 102;
242         mtname = "capture";
243     }
244
245     card_9
246     {
247         mfd = 3;
248         mtd = 251;
249         mtname = "mubar";
250     }
251
252     card_9
253     {
254         mfd = 3;
255         mtd = 252;
256         mtname = "xi";
257     }
258
259     card_9
260     {
261         mfd = 3;
262         mtd = 253;
263         mtname = "gamma";
264     }
265
266     card_9
267     {
268         mfd = 3;
```

```
269         mtd = 259;
270         mtname = "1/v";
271     }
272
273     card_9
274     {
275         mfd = 6;
276         mtd = 2;
277         mtname = "elastic";
278     }
279
280     card_9
281     {
282         mfd = 6;
283         mtd = 16;
284         mtname = "n2n";
285     }
286
287     card_9
288     {
289         mfd = 6;
290         mtd = 17;
291         mtname = "n,3n";
292     }
293
294     card_9
295     {
296         mfd = 6;
297         mtd = 18;
298         mtname = "fission";
299     }
300
301     card_9
302     {
303         mfd = 6;
304         mtd = 51;
305         mtname = "discrete inelastic";
306     }
307
308     card_9
309     {
310         mfd = 6;
311         mtd = -59;
312         mtname = "continued";
313     }
314
315     card_9
316     {
317         mfd = 6;
318         mtd = 91;
319         mtname = "continuum inelastic";
320     }
321
322     /* Terminate temperature 300.0. */
323     card_9
324     {
325         mfd = 0;
326     }
327
328     /* Reactions for temperature 900.0. */
329     card_9
330     {
331         mfd = 3;
332         mtd = 1;
333         mtname = "total";
334     }
335
336     card_9
```

```
337     {
338         mfd = 3;
339         mtd = 2;
340         mtname = "elastic";
341     }
342
343     card_9
344     {
345         mfd = 3;
346         mtd = 18;
347         mtname = "fission";
348     }
349
350     card_9
351     {
352         mfd = 3;
353         mtd = 102;
354         mtname = "capture";
355     }
356
357     card_9
358     {
359         mfd = 6;
360         mtd = 2;
361         mtname = "elastic";
362     }
363
364     /* Terminate temperature 900.0. */
365     card_9
366     {
367         mfd = 0;
368     }
369
370     /* Reactions for temperature 2100.0. */
371     card_9
372     {
373         mfd = 3;
374         mtd = 1;
375         mtname = "total";
376     }
377
378     card_9
379     {
380         mfd = 3;
381         mtd = 2;
382         mtname = "elastic";
383     }
384
385     card_9
386     {
387         mfd = 3;
388         mtd = 18;
389         mtname = "fission";
390     }
391
392     card_9
393     {
394         mfd = 3;
395         mtd = 102;
396         mtname = "capture";
397     }
398
399     card_9
400     {
401         mfd = 6;
402         mtd = 2;
403         mtname = "elastic";
404     }
```

```
405
406      /* Terminate temperature 2100.0. */
407      card_9
408      {
409          mfd = 0;
410      }
411
412      /* Terminate groupr. */
413      card_10
414      {
415          matd = 0;
416      }
417  }
418
419  ccccr
420  {
421      card_1
422      {
423          nin = -25;
424          nisot = 26;
425          nbrks = 27;
426          ndlay = 0; // dlayxs not wanted
427      }
428
429      card_2
430      {
431          lprint = 1;
432          ivers = 1;
433          huse = "t2lanl njoy";
434      }
435
436      card_3
437      {
438          /* hsetid does not have to be 12 chars? */
439          hsetid = "ccccr tests for njoy87";
440      }
441
442      card_4
443      {
444          ngroup = 50;
445          nggrup = 0;
446          niso = 1; // Denotes number of card_5's.
447          maxord = 4;
448          ifopt = 1; // Blocking by reaction order.
449      }
450
451      card_5
452      {
453          /* Note that the original input does not denote the first four
454             variables as strings.
455             What does the two ' denote? Seems a bit irregular.
456           */
457          hisnm = "pu238";
458          habsid = "pu238";
459          hident = "endfb4";
460          hmat = "1050";
461          imat = 1050;
462          xspo = 10.89;
463      }
464
465      card_1
466      {
467          nsblok = 1;
468          maxup = 0; // Always zero (?).
469          maxdn = 50;
470          ichix = -1; // Vector (using groupr flux).
471      }
472
```

```
    card_4
    {
        kbr = 0;
        amass = 2.3821e02;
        efiss = 3.3003e-11;
        ecapt = 1.7461e-12;
        temp = 0.0;
        sigpot = 1.0e10;
        adens = 0.0;
    }

    card_1
    {
        nti = 3;
        nzi = 6;
    }

    card_2
    {
        /* Number of expected temperatures defined by nti. */
        atem[0] = 300;
        atem[1] = 900;
        atem[2] = 2100;
    }

    card_3
    {
        /* Number of expected sigpo values defined by nzi. */
        asig[0] = 1.0e5;
        asig[1] = 1.0e4;
        asig[2] = 1000.0;
        asig[3] = 100.0;
        asig[4] = 10.0;
        asig[5] = 1;
    }
}

moder
{
    card_1
    {
        nin = -24;
        nout = 28;
    }
}
```

Expected NJOY Input Instructions for Test Problem 02

```
moder
20 -21/ ### card_1
reconr
-21 -22/ ### card_1
'pendf tape for pu-238 from endf/b-iv tape 404'/ ### card_2
1050 3/ ### card_3
0.005/ ### card_4
'94-pu-238 from endf/b tape t404'/ ### card_5
'processed by the njoy nuclear data processing system'/ ### card_5
'see original endf/b-iv tape for details of evaluation'/ ### card_5
0/ ### card_3
broadr
-21 -22 -23/ ### card_1
1050 3 0 1 0/ ### card_2
0.005/ ### card_3
300.0 900.0 2100.0/ ### card_4
0/ ### card_5
moder
-23 33/ ### card_1
```

```
unresr
-21 -23 -24/ ### card_1
1050 3 7 1/ ### card_2
300 900 2100/ ### card_3
1.0e10 1.0e5 1.0e4 1000.0 100.0 10.0 1/ ### card_4
0/ ### card_2
groupr
-21 -24 0 -25/ ### card_1
1050 5 0 4 3 3 7 1/ ### card_2
'94-pu-238'/ ### card_3
300.0 900.0 2100.0/ ### card_4
1.0e10 1.0e5 1.0e4 1000.0 100.0 10.0 1/ ### card_5
0.1 0.025 0.8208e06 1.4e06/ ### card_8c
3 1 'total'/ ### card_9
3 2 'elastic'/ ### card_9
3 16 'n2n'/ ### card_9
3 17 'n3n'/ ### card_9
3 18 'fission'/ ### card_9
3 102 'capture'/ ### card_9
3 251 'mubar'/ ### card_9
3 252 'xi'/ ### card_9
3 253 'gamma'/ ### card_9
3 259 '1/v'/ ### card_9
6 2 'elastic'/ ### card_9
6 16 'n2n'/ ### card_9
6 17 'n,3n'/ ### card_9
6 18 'fission'/ ### card_9
6 51 'discrete inelastic'/ ### card_9
6 -59 'continued'/ ### card_9
6 91 'continuum inelastic'/ ### card_9
0/ ### card_9
3 1 'total'/ ### card_9
3 2 'elastic'/ ### card_9
3 18 'fission'/ ### card_9
3 102 'capture'/ ### card_9
6 2 'elastic'/ ### card_9
0/ ### card_9
3 1 'total'/ ### card_9
3 2 'elastic'/ ### card_9
3 18 'fission'/ ### card_9
3 102 'capture'/ ### card_9
6 2 'elastic'/ ### card_9
0/ ### card_9
0/ ### card_10
ccccr
-25 26 27 0/ ### card_1
1 1 't2lanl njoy'/ ### card_2
'ccccr tests for njoy87'/ ### card_3
50 0 1 4 1/ ### card_4
'pu238' 'pu238' 'endfb4' '1050' 1050 10.89/ ### card_5
1 0 50 -1/ ### card_1
0 2.3821e02 3.3003e-11 1.7461e-12 0.0 1.0e10 0.0/ ### card_4
3 6/ ### card_1
300 900 2100/ ### card_2
1.0e5 1.0e4 1000.0 100.0 10.0 1/ ### card_3
moder
-24 28/ ### card_1
stop
```

B.3 Test Problem 03 (tp03)

NIF Version of Test Problem 03

```
reconr
{
    card_1
    {
        nendf = 30;
        npend = 31;
    }

    card_2
    {
        tlabel = "pendf tape for photon interaction cross sections from dlc7e
             ";
    }

    card_3
    {
        mat = 1;
        ncards = 1;
        ngrid = 0;
    }

    card_4
    {
        err = 0.001; // Note the C-style float format with preceding 0.
    }

    card_5
    {
        cards = "1-hydrogen";
    }

    card_3
    {
        mat = 92;
        ncards = 1;
        ngrid = 0;
    }

    card_4
    {
        err = 0.001; // Note the C-style float format with preceding 0.
    }

    card_5
    {
        cards = "92-uranium";
    }

    card_3
    {
        mat = 0;
    }
}

gaminr
{
    card_1
    {
        nendf = 32;
        npend = 31;
        ngam1 = 0;
        ngam2 = 33;
    }

```

```
64      card_2
65      {
66          matb = 1;
67          igg = 3;
68          iwt = 3;
69          lord = 4;
70          iprint = 1;
71      }
72
73      card_3
74      {
75          title = "12 group photon interaction library";
76      }
77
78      card_6
79      {
80          mfd = -1;
81          mtd = 0;
82      }
83
84      card_7
85      {
86          matd = 92;
87      }
88
89      card_6
90      {
91          mfd = -1;
92          mtd = 0;
93      }
94
95      card_7
96      {
97          matd = 0;
98      }
99  }
100
101 dtfr
102 {
103     card_1
104     {
105         nin = 33;
106         nout = 34;
107         npend = 31;
108         nplot = 36;
109     }
110
111     card_2
112     {
113         iprint = 1;
114         ifilm = 1;
115         iedit = 0;
116     }
117
118     card_3
119     {
120         nlmax = 5;
121         ng = 12;
122         iptotl = 4;
123         ipingp = 5;
124         itabl = 16;
125         ned = 1;
126         ntherm = 0;
127     }
128
129     card_4
130     {
131         /* iptotl-3 names will be read, i.e. 4-3 = 1 in this case. */
```

```
132           edits[0] = "pheat";
133       }
134
135       card_5
136       {
137           /* ned triplets, i.e. 1 triplet in this case. */
138           jpos[0] = 1;
139           mt[0] = 621;
140           mult[0] = 1;
141       }
142
143       card_7
144       {
145           nptabl = 0;
146       }
147
148       /* One card_8 for each table set desired. Empty card denotes termination
149          of dtfr.
150        */
151       card_8
152       {
153           hisnam = "h";
154           mat = 1;
155           jsigz = 1;
156           dtemp = 0.0;
157       }
158
159       card_8
160       {
161           hisnam = "u";
162           mat = 92;
163           jsigz = 1;
164           dtemp = 0.0;
165       }
166
167       card_8 {} // Terminate dtfr.
168   }
169
170   matxsr
171   {
172       card_1
173       {
174           ngen1 = 0;
175           ngen2 = 33;
176           nmatx = 35;
177       }
178
179       card_2
180       {
181           ivers = 1;
182           huse = "t2lanl njoy";
183       }
184
185       card_3
186       {
187           npart = 1;
188           ntype = 1;
189           nholl = 1;
190           nmat = 2;
191       }
192
193       card_4
194       {
195           hsetid = "12-group photon interaction library";
196       }
197
198       card_5
199       {
```

```
200           hpart = "g";
201       }
202
203       card_6
204       {
205           ngrp = 12;
206       }
207
208       card_7
209       {
210           htype = "gscat";
211       }
212
213       card_8
214       {
215           jinp = 1;
216       }
217
218       card_9
219       {
220           joutp = 1;
221       }
222
223       /* One card_10 per material. */
224       card_10
225       {
226           hmat = "h";
227           matno = 1;
228           matgg = 1;
229       }
230
231       card_10
232       {
233           hmat = "u";
234           matno = 92;
235           matgg = 92;
236       }
237   }
238
239   viewr
240   {
241       /* Documentation names the first two cards as card 1. Use card 0 to
               denote
242          the first card, just like in plotr.
243        */
244       card_0
245       {
246           infile = 36;
247           nps = 37;
248       }
249   }
```

Expected NJOY Input Instructions for Test Problem 03

```
1   reconr
2   30 31/ ### card_1
3   'pendf tape for photon interaction cross sections from dlc7e'/ ### card_2
4   1 1 0/ ### card_3
5   0.001/ ### card_4
6   '1-hydrogen'/ ### card_5
7   92 1 0/ ### card_3
8   0.001/ ### card_4
9   '92-uranium'/ ### card_5
10  0/ ### card_3
11  gaminr
12  32 31 0 33/ ### card_1
13  1 3 3 4 1/ ### card_2
```

```
'12 group photon interaction library'/ ### card_3
-1 0/ ### card_6
92/ ### card_7
-1 0/ ### card_6
0/ ### card_7
dtfr
33 34 31 36/ ### card_1
1 1 0/ ### card_2
5 12 4 5 16 1 0/ ### card_3
'pheat'/ ### card_4
1 621 1/ ### card_5
0/ ### card_7
'h' 1 1 0.0/ ### card_8
'u' 92 1 0.0/ ### card_8
/ ### card_8
matxsr
0 33 35/ ### card_1
1 't2lanl njoy'/ ### card_2
1 1 1 2/ ### card_3
'12-group photon interaction library'/ ### card_4
'g'/ ### card_5
12/ ### card_6
'gscat'/ ### card_7
1/ ### card_8
1/ ### card_9
'h' 1 1/ ### card_10
'u' 92 92/ ### card_10
viewr
36 37/ ### card_0
stop
```

B.4 Test Problem 04 (tp04)

NIF Version of Test Problem 04

```
moder
{
    card_1
    {
        nin = 20;
        nout = -21;
    }
}

reconr
{
    card_1
    {
        nendf = -21;
        npend = -22;
    }

    card_2
    {
        tlabel =  "u-235 10% pendf for errorr test problem from t511";
    }

    card_3
    {
        mat = 1395;
    }

    card_4
    {
        err = 0.10; // Use C-style floats.
    }

    card_3
    {
        mat = 0;
    }
}

errorr
{
    card_1
    {
        nendf = -21;
        npend = -22;
        ngout = 0;
        nout = 23;
        nin = 0;
    }

    card_2
    {
        matd = 1395;
        ign = 19;
        iwt = 3;
        iprint = 1;
        irelco = 1;
    }

    card_3
    {
        mprint = 0;
        tempin = 0;
    }

```

```
65      /* Test problem 04 is using a file of the endf-5 format (iverf = 5) */
66
67      card_7
68      {
69          iread = 0;
70          mfcov = 33;
71      }
72
73      card_12a
74      {
75          ngn = 1;
76      }
77
78      card_12b
79      {
80          egn[0] = 1.0e0;
81          egn[1] = 1.0e3;
82      }
83  }
84
85  groupr
86  {
87      card_1
88      {
89          nendf = -21;
90          npend = -22;
91          ngout1 = 0;
92          ngout2 = 24;
93      }
94
95      card_2
96      {
97          matb = 1395;
98          ign = 3;
99          igg = 0;
100         iwt = 3;
101         lord = 0;
102         ntemp = 1;
103         nsigz = 1;
104         iprint = 1;
105     }
106
107     card_3
108     {
109         title = "u-235 multigroup nubar calculation";
110     }
111
112     card_4
113     {
114         temp[0] = 0.0;
115     }
116
117     card_5
118     {
119         sigz[0] = 1.0e10;
120     }
121
122     card_9
123     {
124         mfd = 3;
125         mtd = 452;
126         mtname = "total nubar";
127     }
128
129     /* Terminate temperature/material with mfd = 0 as usual. */
130     card_9
131     {
132         mfd = 0;
```

```
133     }
134
135     /* Terminate groupr run with matd = 0 as usual. */
136     card_10
137     {
138         matd = 0;
139     }
140 }
141
142 errorr
143 {
144     card_1
145     {
146         nendf = -21;
147         npend = 0;
148         ngout = 24;
149         nout = 25;
150         nin = 23;
151     }
152
153     card_2
154     {
155         matd = 1395;
156         ign = 1;
157         iwt = 2;
158         iprint = 1;
159         irelco = 1;
160     }
161
162     /* Card 3 omitted since ngout != 0. */
163
164     /* Test problem 04 is using a file of the endf-5 format (iverf = 5) */
165
166     card_7
167     {
168         iread = 0;
169         mfcov = 31;
170     }
171
172     card_12a
173     {
174         ngn = 7;
175     }
176
177     card_12b
178     {
179         egn[0] = 1.0e0;
180         egn[1] = 1.0e1;
181         egn[2] = 1.0e2;
182         egn[3] = 1.0e3;
183         egn[4] = 1.0e4;
184         egn[5] = 1.0e5;
185         egn[6] = 1.0e6;
186         egn[7] = 1.0e7;
187     }
188 }
```

Expected NJOY Input Instructions for Test Problem 04

```
1 moder
2 20 -21/ ### card_1
3 reconr
4 -21 -22/ ### card_1
5 'u-235 10% pendf for errorr test problem from t511'/ ### card_2
6 1395/ ### card_3
7 0.10/ ### card_4
8 0/ ### card_3
```

```
 9  errorr
10  -21 -22 0 23 0/ ### card_1
11  1395 19 3 1 1/ ### card_2
12  0 0/ ### card_3
13  0 33/ ### card_7
14  1/ ### card_12a
15  1.0e0 1.0e3/ ### card_12b
16  groupr
17  -21 -22 0 24/ ### card_1
18  1395 3 0 3 0 1 1 1/ ### card_2
19  'u-235 multigroup nubar calculation'/ ### card_3
20  0.0/ ### card_4
21  1.0e10/ ### card_5
22  3 452 'total nubar'/ ### card_9
23  0/ ### card_9
24  0/ ### card_10
25  errorr
26  -21 0 24 25 23/ ### card_1
27  1395 1 2 1 1/ ### card_2
28  0 31/ ### card_7
29  7/ ### card_12a
30  1.0e0 1.0e1 1.0e2 1.0e3 1.0e4 1.0e5 1.0e6 1.0e7/ ### card_12b
31  stop
```

B.5 Test Problem 05 (tp05)

NIF Version of Test Problem 05

```
moder
{
    card_1
    {
        nin = 30;
        nout = -31;
    }
}

moder
{
    card_1
    {
        nin = -31;
        nout = -32;
    }
}

errorr
{
    card_1
    {
        nendf = -31;
        npend = -32;
        ngout = 0;
        nout = -33;
    }

    card_2
    {
        matd = 1306;
        ign = 19;
        iwt = 2;
        iprint = 1;
    }

    card_3
    {
        mprint = 0;
        tempin = 0;
    }

    /* Test problem 05 is using a file of the endf-5 format (iverf=5) */

    card_7
    {
        iread = 0;
        mfcov = 33;
    }

    card_12a
    {
        ngn = 1;
    }

    card_12b
    {
        egn = 1e-5;
        egn = 2e7;
    }
}

covr
{
```

```
65      card_1
66      {
67          nin = -33;
68          nout = 0;
69          nplot = 34;
70      }
71
72      card_2
73      {
74          icolor = 1;
75      }
76
77      card_2a
78      {
79      }
80
81      card_3a
82      {
83      }
84
85      card_4
86      {
87          mat = 1306;
88      }
89  }
90
91  viewr
92  {
93      /* Documentation names the first two cards as card 1. Use card 0 to
            denote
94         the first card, just like in plotr.
95       */
96      card_0
97      {
98          infile = 34;
99          nps = 35;
100     }
101 }
```

Expected NJOY Input Instructions for Test Problem 05

```
1   moder
2   30 -31/ ### card_1
3   moder
4   -31 -32/ ### card_1
5   errorr
6   -31 -32 0 -33/ ### card_1
7   1306 19 2 1/ ### card_2
8   0 0/ ### card_3
9   0 33/ ### card_7
10  1/ ### card_12a
11  1e-5 2e7/ ### card_12b
12  covr
13  -33 0 34/ ### card_1
14  1/ ### card_2
15  / ### card_2a
16  / ### card_3a
17  1306/ ### card_4
18  viewr
19  34 35/ ### card_0
20  stop
```

B.6 Test Problem 06 (tp06)

NIF Version of Test Problem 06

```
plotr
{
    card_0
    {
        nplt = 31;
    }

    card_1 {}

    /* New axes, new page. */
    card_2
    {
        iplot = 1;
    }

    card_3
    {
        /* e should be delimited by < >? Oh well. */
        t1 = "<endf/b-v carbon";
    }

    card_3a
    {
        t2 = "<t>otal <c>ross <s>ection";
    }

    card_4
    {
        itype = 4;
    }

    card_5
    {
        el = 1e3;
        eh = 2e7;
    }

    card_5a {}

    card_6
    {
        yl = 0.5;
        yh = 10;
    }

    card_6a {}

    /* card_7 and card_7a skipped since jtype = 0. */

    card_8
    {
        iverf = 5;
        nin = 30;
        matd = 1306;
        mfd = 3;
        mtd = 1;
    }

    /* card_9 since it's a 2d plot (indicated by sign of itype in card_4) */
    card_9 {}

    /* New axes, new page. */
    card_2
    {
```

```
65          iplot = 1;
66      }
67
68      card_3
69      {
70          /* e should be delimited by < >? Oh well. */
71          t1 = "<endf/b-v carbon";
72      }
73
74      card_3a
75      {
76          t2 = "(n,]a>) with fake data";
77      }
78
79      card_4
80      {
81          itype = 1;
82          jtype = 0;
83          igrid = 2;
84          ileg = 1;
85          xtag = 1.3e7;
86          ytag = 0.32;
87      }
88
89      card_5 {}
90      card_5a {}
91      card_6 {}
92      card_6a {}
93      /* card_7 and card_7a skipped since jtype = 0 */
94
95      card_8
96      {
97          iverf = 5;
98          nin = 30;
99          matd = 1306;
100         mfd = 3;
101         mtd = 107;
102     }
103
104     card_9 {}
105
106     card_10
107     {
108         aleg = "<endf/b-v mat1306";
109     }
110
111     /* Add plot on existing axes. */
112     card_2
113     {
114         iplot = 2;
115     }
116
117     /* card 3-7 skipped since iplot = 2. */
118
119     card_8
120     {
121         iverf = 0; // Ignore rest of parameters on card.
122     }
123
124     card_9
125     {
126         icon = -1;
127         isym = 0;
128     }
129
130     /* card_10 since ileg = 1. */
131     card_10
132     {
```

```
133         aleg = "<s>mith & <s>mith 1914";
134     }
135
136     /* card_12 since iverf = 0. */
137     card_12
138     {
139         nform = 0;
140     }
141
142     /* card_13 since nform = 0. */
143     card_13
144     {
145         xdata = 1.1e7;
146         ydata = 0.08;
147         yerr1 = 0.05;
148         yerr2 = 0.05;
149     }
150
151     card_13
152     {
153         xdata = 1.2e7;
154         ydata = 0.10;
155         yerr1 = 0.05;
156         yerr2 = 0.05;
157     }
158
159     card_13
160     {
161         xdata = 1.3e7;
162         ydata = 0.09;
163         yerr1 = 0.04;
164         yerr2 = 0.04;
165     }
166
167     card_13
168     {
169         xdata = 1.4e7;
170         ydata = 0.08;
171         yerr1 = 0.03;
172         yerr2 = 0.03;
173     }
174
175     /* Terminate card_13 with empty card. */
176     card_13 {}
177
178     /* Add plot on existing axes. */
179     card_2
180     {
181         iplot = 3;
182     }
183
184     /* Card 3-7 skipped since iplot = 3. */
185
186     card_8
187     {
188         iverf = 0; // Ignore rest of parameters on card.
189     }
190
191     card_9
192     {
193         icon = -1;
194         isym = 2;
195     }
196
197     /* card_10 since ileg = 1. */
198     card_10
199     {
200         aleg = "<b>lack & <b>lue 2008";
```

```
201     }
202
203     /* card_12 since iverf = 0. */
204     card_12
205     {
206         nform = 0;
207     }
208
209     /* card_13 since nform = 0. */
210     card_13
211     {
212         xdata = 1.15e7;
213         ydata = 0.07;
214         yerr1 = 0.02;
215         yerr2 = 0.0;
216         xerr1 = 0.2e6;
217         xerr2 = 0.0;
218     }
219
220     card_13
221     {
222         xdata = 1.25e7;
223         ydata = 0.11;
224         yerr1 = 0.02;
225         yerr2 = 0.0;
226         xerr1 = 0.2e6;
227         xerr2 = 0.0;
228     }
229
230     card_13
231     {
232         xdata = 1.35e7;
233         ydata = 0.08;
234         yerr1 = 0.015;
235         yerr2 = 0.0;
236         xerr1 = 0.2e6;
237         xerr2 = 0.0;
238     }
239
240     card_13
241     {
242         xdata = 1.45e7;
243         ydata = 0.075;
244         yerr1 = 0.01;
245         yerr2 = 0.0;
246         xerr1 = 0.2e6;
247         xerr2 = 0.0;
248     }
249
250     /* Terminate card_13 with empty card. */
251     card_13 {}
252
253     /* New axes, new page. */
254     card_2
255     {
256         iplot = 1;
257     }
258
259     card_3
260     {
261         /* e should be delimited by < >? Oh well. */
262         t1 = "<endf/b-v carbon";
263     }
264
265     card_3a
266     {
267         t2 = "<e>lastic <mf4>";
268     }
```

```
269
270       card_4
271       {
272           itype = -1; // 3d axes.
273           jtype = 2;
274       }
275
276       card_5 {}
277       card_5a {}
278       card_6 {}
279       card_6a {}
280       card_7 {}
281       card_7a {}
282
283       card_8
284       {
285           iverf = 5;
286           nin = 30;
287           matd = 1306;
288           mfd = 4;
289           mtd = 2;
290       }
291
292       card_11 {}
293
294       /* New axes, new page. */
295       card_2
296       {
297           iplot = 1;
298       }
299
300       card_3
301       {
302           t1 = "<endf/b-v l>i-6";
303       }
304
305       card_3a
306       {
307           t2 = "(n,2n)]a >neutron distribution";
308       }
309
310       card_4
311       {
312           itype = -1;
313           jtype = 2;
314       }
315
316       card_5 {}
317       card_5a {}
318
319       card_6
320       {
321           yl = 0;
322           yh = 12e6;
323           ystep = 2e6;
324       }
325
326       card_6a {}
327       card_7 {}
328       card_7a {}
329
330       card_8
331       {
332           iverf = 5;
333           nin = 30;
334           matd = 1303;
335           mfd = 5;
336           mtd = 24;
```

```
}

/* 3D plot. */
card_11 {}

/* New axes, new page. */
card_2
{
    iplot = 1;
}

card_3
{
    t1 = "<endf/b-v l>i-6";
}

card_3a
{
    t2 = "(n,2n)]a >neutron spectra vs <E>";
}

card_4
{
    itype = 4;
    jtype = 0;
    igrid = 2;
    ileg = 2;
}

card_5
{
    el = 10.0;
    eh = 2.0e7;
}

card_5a {}

card_6
{
    yl = 1e-11;
    yh = 1e-6;
}

card_6a
{
    ylabl = "<c>ross <s>ection (barns/e<v>)";
}

card_8
{
    iverf = 5;
    nin = 30;
    matd = 1303;
    mfd = 5;
    mtd = 24;
    temper = 0.0;
    nth = 12;
}

card_9 {}

card_10
{
    aleg = "10 <m>e<v";
}

card_10a
{
```

```
405         xtag = 1e3;
406         ytag = 2e-11;
407         xpoint = 1e2;
408     }
409
410     /* 2th additional plot on existing axes. */
411     card_2
412     {
413         iplot = 2;
414     }
415
416     card_8
417     {
418         iverf = 5;
419         nin = 30;
420         matd = 1303;
421         mfd = 5;
422         mtd = 24;
423         temper = 0.0;
424         nth = 16;
425     }
426
427     card_9 {}
428
429     /* card 10, 10a since ileg = 2 for the current axes. */
430     card_10
431     {
432         aleg = "14 <m>e<v";
433     }
434
435     card_10a
436     {
437         xtag = 1e4;
438         ytag = 2e-10;
439         xpoint = 2e3;
440     }
441
442     /* 3rd additional plot on existing axes. */
443     card_2
444     {
445         iplot = 3;
446     }
447
448     card_8
449     {
450         iverf = 5;
451         nin = 30;
452         matd = 1303;
453         mfd = 5;
454         mtd = 24;
455         temper = 0.0;
456         nth = 20;
457     }
458
459     card_9 {}
460
461     card_10
462     {
463         aleg = "20 <m>e<v";
464     }
465
466     card_10a
467     {
468         xtag = 1e5;
469         ytag = 2e-9;
470         xpoint = 4e4;
471     }
472
```

```
473      /* Terminate plotting job. */
474      card_2
475      {
476          iplot = 99;
477      }
478  }
479
480  viewr
481  {
482      /* Documentation names the first two cards as card 1. Use card 0 to
             denote
483         the first card, just like in plotr.
484       */
485      card_0
486      {
487          infile = 31;
488          nps = 32;
489      }
490  }
```

Expected NJOY Input Instructions for Test Problem 06

```
1   plotr
2   31/ ### card_0
3   / ### card_1
4   1/ ### card_2
5   '<endf/b-v carbon'/ ### card_3
6   '<t>otal <c>ross <s>ection'/ ### card_3a
7   4/ ### card_4
8   1e3 2e7/ ### card_5
9   / ### card_5a
10  0.5 10/ ### card_6
11  / ### card_6a
12  5 30 1306 3 1/ ### card_8
13  / ### card_9
14  1/ ### card_2
15  '<endf/b-v carbon'/ ### card_3
16  '(n,]a>) with fake data'/ ### card_3a
17  1 0 2 1 1.3e7 0.32/ ### card_4
18  / ### card_5
19  / ### card_5a
20  / ### card_6
21  / ### card_6a
22  5 30 1306 3 107/ ### card_8
23  / ### card_9
24  '<endf/b-v mat1306'/ ### card_10
25  2/ ### card_2
26  0/ ### card_8
27  -1 0/ ### card_9
28  '<s>mith & <s>mith 1914'/ ### card_10
29  0/ ### card_12
30  1.1e7 0.08 0.05 0.05/ ### card_13
31  1.2e7 0.10 0.05 0.05/ ### card_13
32  1.3e7 0.09 0.04 0.04/ ### card_13
33  1.4e7 0.08 0.03 0.03/ ### card_13
34  / ### card_13
35  3/ ### card_2
36  0/ ### card_8
37  -1 2/ ### card_9
38  '<b>lack & <b>lue 2008'/ ### card_10
39  0/ ### card_12
40  1.15e7 0.07 0.02 0.0 0.2e6 0.0/ ### card_13
41  1.25e7 0.11 0.02 0.0 0.2e6 0.0/ ### card_13
42  1.35e7 0.08 0.015 0.0 0.2e6 0.0/ ### card_13
43  1.45e7 0.075 0.01 0.0 0.2e6 0.0/ ### card_13
44  / ### card_13
45  1/ ### card_2
```

```
'<endf/b-v carbon'/ ### card_3
'<e>lastic <mf4>'/ ### card_3a
-1 2/ ### card_4
/ ### card_5
/ ### card_5a
/ ### card_6
/ ### card_6a
/ ### card_7
/ ### card_7a
5 30 1306 4 2/ ### card_8
/ ### card_11
1/ ### card_2
'<endf/b-v l>i-6'/ ### card_3
'(n,2n)]a >neutron distribution'/ ### card_3a
-1 2/ ### card_4
/ ### card_5
/ ### card_5a
0 12e6 2e6/ ### card_6
/ ### card_6a
/ ### card_7
/ ### card_7a
5 30 1303 5 24/ ### card_8
/ ### card_11
1/ ### card_2
'<endf/b-v l>i-6'/ ### card_3
'(n,2n)]a >neutron spectra vs <E>'/ ### card_3a
4 0 2 2/ ### card_4
10.0 2.0e7/ ### card_5
/ ### card_5a
1e-11 1e-6/ ### card_6
'<c>ross <s>ection (barns/e<v>)'/ ### card_6a
5 30 1303 5 24 0.0 12/ ### card_8
/ ### card_9
'10 <m>e<v'/ ### card_10
1e3 2e-11 1e2/ ### card_10a
2/ ### card_2
5 30 1303 5 24 0.0 16/ ### card_8
/ ### card_9
'14 <m>e<v'/ ### card_10
1e4 2e-10 2e3/ ### card_10a
3/ ### card_2
5 30 1303 5 24 0.0 20/ ### card_8
/ ### card_9
'20 <m>e<v'/ ### card_10
1e5 2e-9 4e4/ ### card_10a
99/ ### card_2
viewr
31 32/ ### card_0
stop
```

B.7 Test Problem 07 (tp07)

NIF Version of Test Problem 07

```
moder
{
    card_1
    {
        nin = 20;
        nout = -21;
    }
}

reconr
{
    card_1
    {
        nendf = -21;
        npend = -22;
    }

    card_2
    {
        tlabel = "pendf tape for u-235 from endf/b-v tape 511";
    }

    card_3
    {
        mat = 1395;
        ncards = 3;
    }

    card_4
    {
        /* Note C-style float compared to the original declaration above. */
        err = 0.005;
    }

    card_5
    {
        cards = "92-u-235 from endf/b-v tape 511 ";
    }

    card_5
    {
        cards = "processed by the njoy nuclear data processing system";
    }

    card_5
    {
        cards = "see original endf/b-v tape for details of evaluation";
    }

    /* Terminate execution of reconr with mat = 0 as usual. */
    card_3
    {
        mat = 0;
    }
}

broadr
{
    card_1
    {
        nendf = -21;
        nin = -22;
        nout = -23;
    }
```

```
65
66      card_2
67      {
68          mat1 = 1395;
69          ntemp2 = 1;
70          istart = 0;
71          istrap = 1;
72          temp1 = 0;
73      }
74
75      card_3
76      {
77          errthn = 0.005;
78      }
79
80      card_4
81      {
82          temp2[0] = 300;
83      }
84
85      /* Terminate execution of broadr with mat1 = 0 as usual. */
86      card_5
87      {
88          mat1 = 0;
89      }
90  }
91
92  heatr
93  {
94      card_1
95      {
96          nendf = -21;
97          nin = -23;
98          nout = -24;
99          /* nplot not supplied, defaulted to 0? */
100     }
101
102     card_2
103     {
104         matd = 1395;
105     }
106 }
107
108 moder
109 {
110     card_1
111     {
112         nin = -24;
113         nout = 28;
114     }
115 }
116
117 groupr
118 {
119     card_1
120     {
121         nendf = -21;
122         npend = -24;
123         ngout1 = 0;
124         ngout2 = -25;
125     }
126
127     card_2
128     {
129         matb = 1395;
130         ign = 3;
131         igg = 2;
132         iwt = 9;
```

```
133          lord = 0;
134          ntemp = 1;
135          nsigz = 1;
136          iprint = 1;
137      }
138
139      card_3
140      {
141          title = "u-235 from tape 511";
142      }
143
144      card_4
145      {
146          temp[0] = 300.0;
147      }
148
149      card_5
150      {
151          sigz[0] = 1.0e10;
152      }
153
154      card_9
155      {
156          mfd = 16;
157          /* mtd and mtname does not have to be supplied? */
158      }
159
160      /* Terminate temperature/material with mfd = 0 as usual. */
161      card_9
162      {
163          mfd = 0;
164      }
165
166      /* Terminate groupr run with matd = 0 as usual. */
167      card_10
168      {
169          matd = 0;
170      }
171  }
172
173  acer
174  {
175      card_1
176      {
177          nendf = -21;
178          npend = -24;
179          ngend = -25;
180          nace = 26;
181          ndir = 27;
182      }
183
184      card_2
185      {
186          iopt = 1;
187      }
188
189      card_3
190      {
191          hk = "njoy test problem 7";
192      }
193
194      card_5
195      {
196          matd = 1395;
197          tempd = 300.0;
198      }
199
200      card_6
```

```
    {
        newfor = 0;
    }

    card_7 {}
}
```

Expected NJOY Input Instructions for Test Problem 07

```
moder
20 -21/ ### card_1
reconr
-21 -22/ ### card_1
'pendf tape for u-235 from endf/b-v tape 511'/ ### card_2
1395 3/ ### card_3
0.005/ ### card_4
'92-u-235 from endf/b-v tape 511 '/ ### card_5
'processed by the njoy nuclear data processing system'/ ### card_5
'see original endf/b-v tape for details of evaluation'/ ### card_5
0/ ### card_3
broadr
-21 -22 -23/ ### card_1
1395 1 0 1 0/ ### card_2
0.005/ ### card_3
300/ ### card_4
0/ ### card_5
heatr
-21 -23 -24/ ### card_1
1395/ ### card_2
moder
-24 28/ ### card_1
groupr
-21 -24 0 -25/ ### card_1
1395 3 2 9 0 1 1 1/ ### card_2
'u-235 from tape 511'/ ### card_3
300.0/ ### card_4
1.0e10/ ### card_5
16/ ### card_9
0/ ### card_9
0/ ### card_10
acer
-21 -24 -25 26 27/ ### card_1
1/ ### card_2
'njoy test problem 7'/ ### card_3
1395 300.0/ ### card_5
0/ ### card_6
/ ### card_7
stop
```

B.8 Test Problem 08 (tp08)

NIF Version of Test Problem 08

```
moder
{
    card_1
    {
        nin = 20;
        nout = -21;
    }
}

reconr
{
    card_1
    {
        nendf = -21;
        npend = -22;
    }

    card_2
    {
        tlabel = "pendf tape for endf/b-vi.1 28-ni-61a";
    }

    card_3
    {
        mat = 2834;
        ncards = 1;
        ngrid = 0;
    }

    card_4
    {
        /* Note C-style float compared to the original declaration above. */
        err = 0.01;
    }

    card_5
    {
        cards = "28-ni-61a from endf/b-vi.1 t124 (hetrick,fu;ornl)";
    }

    /* Terminate execution of reconr with mat = 0 as usual. */
    card_3
    {
        mat = 0;
    }
}

broadr
{
    card_1
    {
        nendf = -21;
        nin = -22;
        nout = -23;
    }

    card_2
    {
        mat1 = 2834;
        ntemp2 = 1;
    }

    card_3
    {
```

```
        errthn = 0.01;
    }

    card_4
    {
        temp2[0] = 300;
    }

    /* Terminate execution of broadr with mat1 = 0 as usual. */
    card_5
    {
        mat1 = 0;
    }
}

heatr
{
    card_1
    {
        nendf = -21;
        nin = -23;
        nout = -24;
        /* nplot not supplied, defaulted to 0? */
    }

    card_2
    {
        matd = 2834;
        npk = 6;
        nqa = 0;
        ntemp = 1;
        local = 0;
        iprint = 2;
    }

    card_3
    {
        mtk[0] = 302;
        mtk[1] = 303;
        mtk[2] = 304;
        mtk[3] = 402;
        mtk[4] = 443;
        mtk[5] = 444;
    }
}

moder
{
    card_1
    {
        nin = -24;
        nout = 28;
    }
}

groupr
{
    card_1
    {
        nendf = -21;
        npend = -24;
        ngout1 = 0;
        ngout2 = -22;
    }

    card_2
    {
        matb = 2834;
```

```
133         ign = 3;
134         igg = 3;
135         iwt = 9;
136         lord = 4;
137         ntemp = 1;
138         nsigz = 1;
139         iprint = 1;
140     }
141
142     card_3
143     {
144         title = "ni61a endf/b-vi.1 30x12";
145     }
146
147     card_4
148     {
149         temp[0] = 300;
150     }
151
152     card_5
153     {
154         sigz[0] = 1e10; // No trailing dots. Use C-style floats.
155     }
156
157     card_9
158     {
159         mfd = 3;
160         /* mtd and mtname does not have to be supplied? */
161     }
162
163     card_9
164     {
165         mfd = 3;
166         mtd = 251;
167         mtname = "mubar";
168     }
169
170     card_9
171     {
172         mfd = 3;
173         mtd = 252;
174         mtname = "xi";
175     }
176
177     card_9
178     {
179         mfd = 3;
180         mtd = 253;
181         mtname = "gamma";
182     }
183
184     card_9
185     {
186         mfd = 3;
187         mtd = 259;
188         mtname = "1/v";
189     }
190
191     card_9
192     {
193         mfd = 6;
194         /* mtd and mtname does not have to be supplied? */
195     }
196
197     card_9
198     {
199         mfd = 16;
200         /* mtd and mtname does not have to be supplied? */
```

```
    }

    /* Terminate temperature/material with mfd = 0 as usual. */
    card_9
    {
        mfd = 0;
    }

    /* Terminate groupr run with matd = 0 as usual. */
    card_10
    {
        matd = 0;
    }
}

acer
{
    card_1
    {
        nendf = -21;
        npend = -24;
        ngend = 0;
        nace = 25;
        ndir = 26;
    }

    card_2
    {
        iopt = 1;
        iprint = 1;
        ntype = 1;
    }

    card_3
    {
        hk = "28-ni-61a from endf-vi.1";
    }

    card_5
    {
        matd = 2834;
        tempd = 300.0;
    }

    card_6
    {
        newfor = 0;
    }

    card_7 {}
}
```

Expected NJOY Input Instructions for Test Problem 08

```
moder
20 -21/ ### card_1
reconr
-21 -22/ ### card_1
'pendf tape for endf/b-vi.1 28-ni-61a'/ ### card_2
2834 1 0/ ### card_3
0.01/ ### card_4
'28-ni-61a from endf/b-vi.1 t124 (hetrick,fu;ornl)'/ ### card_5
0/ ### card_3
broadr
-21 -22 -23/ ### card_1
2834 1/ ### card_2
0.01/ ### card_3
```

```
300/ ### card_4
0/ ### card_5
heatr
-21 -23 -24/ ### card_1
2834 6 0 1 0 2/ ### card_2
302 303 304 402 443 444/ ### card_3
moder
-24 28/ ### card_1
groupr
-21 -24 0 -22/ ### card_1
2834 3 3 9 4 1 1 1/ ### card_2
'ni61a endf/b-vi.1 30x12'/ ### card_3
300/ ### card_4
1e10/ ### card_5
3/ ### card_9
3 251 'mubar'/ ### card_9
3 252 'xi'/ ### card_9
3 253 'gamma'/ ### card_9
3 259 '1/v'/ ### card_9
6/ ### card_9
16/ ### card_9
0/ ### card_9
0/ ### card_10
acer
-21 -24 0 25 26/ ### card_1
1 1 1/ ### card_2
'28-ni-61a from endf-vi.1'/ ### card_3
2834 300.0/ ### card_5
0/ ### card_6
/ ### card_7
stop
```

B.9 Test Problem 10 (tp10)

NIF Version of Test Problem 10

```
moder
{
    card_1
    {
        nin = 20;
        nout = -21;
    }
}

reconr
{
    card_1
    {
        nendf = -21;
        npend = -22;
    }

    card_2
    {
        tlabel = "pendf tape for pu-238 from endf/b-iv tape 404";
    }

    card_3
    {
        mat = 1050;
        ncards = 3;
    }

    card_4
    {
        /* Note C-style float compared to the original declaration above. */
        err = 0.005;
    }

    card_5
    {
        cards = "94-pu-238 from endf/b tape t404";
    }

    card_5
    {
        cards = "processed by the njoy nuclear data processing system";
    }

    card_5
    {
        cards = "see original endf/b-iv tape for details of evaluation";
    }

    /* Terminate execution of reconr with mat = 0 as usual. */
    card_3
    {
        mat = 0;
    }
}

broadr
{
    card_1
    {
        nendf = -21;
        nin = -22;
        nout = -23;
    }
```

```
65
66      card_2
67      {
68          mat1 = 1050;
69          ntemp2 = 3;
70          istart = 0;
71          istrap = 1;
72          temp1 = 0;
73      }
74
75      card_3
76      {
77          errthn = 0.005;
78      }
79
80      card_4
81      {
82          temp2[0] = 300.0;
83          temp2[1] = 900.0;
84          temp2[2] = 2100.0;
85      }
86
87      /* Terminate execution of broadr with mat1 = 0 as usual. */
88      card_5
89      {
90          mat1 = 0;
91      }
92  }
93
94  unresr
95  {
96      card_1
97      {
98          nendf = -21;
99          nin = -23;
100         nout = -24;
101     }
102
103     card_2
104     {
105         matd = 1050;
106         ntemp = 3;
107         nsigz = 7;
108         iprint = 1;
109     }
110
111     card_3
112     {
113         temp[0] = 300;
114         temp[1] = 900;
115         temp[2] = 2100;
116     }
117
118     card_4
119     {
120         sigz[0] = 1.0e10;
121         sigz[1] = 1.0e5;
122         sigz[2] = 1.0e4;
123         sigz[3] = 1000.0;
124         sigz[4] = 100.0;
125         sigz[5] = 10.0;
126         sigz[6] = 1;
127     }
128
129     card_2
130     {
131         matd = 0;
132     }
```

```
}

purr
{
    card_1
    {
        nendf = -21;
        nin = -24;
        nout = -25;
    }

    card_2
    {
        matd = 1050;
        ntemp = 3;
        nsigz = 7;
        nbin = 20;
        nladr = 4;
    }

    card_3
    {
        temp[0] = 300;
        temp[1] = 900;
        temp[2] = 2100;
    }

    card_4
    {
        sigz[0] = 1.0e10;
        sigz[1] = 1.0e5;
        sigz[2] = 1.0e4;
        sigz[3] = 1000.0;
        sigz[4] = 100.0;
        sigz[5] = 10.0;
        sigz[6] = 1;
    }

    card_2
    {
        matd = 0;
    }
}

acer
{
    card_1
    {
        nendf = -21;
        npend = -25;
        ngend = 0;
        nace = 26;
        ndir = 27;
    }

    card_2
    {
        iopt = 1;
    }

    card_3
    {
        hk = "njoy test problem 10";
    }

    card_5
    {
        matd = 1050;
```

```
        tempd = 300.0;
    }

    card_6 {}
    card_7 {}
}

moder
{
    card_1
    {
        nin = -25;
        nout = 28;
    }
}
```

Expected NJOY Input Instructions for Test Problem 10

```
moder
20 -21/ ### card_1
reconr
-21 -22/ ### card_1
'pendf tape for pu-238 from endf/b-iv tape 404'/ ### card_2
1050 3/ ### card_3
0.005/ ### card_4
'94-pu-238 from endf/b tape t404'/ ### card_5
'processed by the njoy nuclear data processing system'/ ### card_5
'see original endf/b-iv tape for details of evaluation'/ ### card_5
0/ ### card_3
broadr
-21 -22 -23/ ### card_1
1050 3 0 1 0/ ### card_2
0.005/ ### card_3
300.0 900.0 2100.0/ ### card_4
0/ ### card_5
unresr
-21 -23 -24/ ### card_1
1050 3 7 1/ ### card_2
300 900 2100/ ### card_3
1.0e10 1.0e5 1.0e4 1000.0 100.0 10.0 1/ ### card_4
0/ ### card_2
purr
-21 -24 -25/ ### card_1
1050 3 7 20 4/ ### card_2
300 900 2100/ ### card_3
1.0e10 1.0e5 1.0e4 1000.0 100.0 10.0 1/ ### card_4
0/ ### card_2
acer
-21 -25 0 26 27/ ### card_1
1/ ### card_2
'njoy test problem 10'/ ### card_3
1050 300.0/ ### card_5
/ ### card_6
/ ### card_7
moder
-25 28/ ### card_1
stop
```

B.10 Test Problem 11 (tp11)

NIF Version of Test Problem 11

```
moder
{
    card_1
    {
        nin = 20;
        nout = -21;
    }
}

reconr
{
    card_1
    {
        nendf = -21;
        npend = -22;
    }

    card_2
    {
        tlabel =  "pendf tape for pu-238 from endf/b-iv tape 404";
    }

    card_3
    {
        mat = 1050;
        ncards = 3;
    }

    card_4
    {
        err = 0.005; // Use C-style floats.
    }

    card_5
    {
        cards = "94-pu-238 from endf/b tape t404";
    }

    card_5
    {
        cards = "processed by the njoy nuclear data processing system";
    }

    card_5
    {
        cards = "see original endf/b-iv tape for details of evaluation";
    }

    /* Card 6 skipped since ngrid defaults to 0 in first card 3 */

    /* Terminate reconr. */
    card_3
    {
        mat = 0;
    }
}

broadr
{
    card_1
    {
        nendf = -21;
        nin = -22;
        nout = -23;
```

```
65      }
66
67      card_2
68      {
69          mat1 = 1050;
70          ntemp2 = 3;
71          istart = 0;
72          istrap = 1;
73          temp1 = 0;
74      }
75
76      card_3
77      {
78          errthn = 0.005; // Use C-style floats.
79      }
80
81      card_4
82      {
83          temp2[0] = 300.0; // Use C-style floats.
84          temp2[1] = 900.0;
85          temp2[2] = 2100.0;
86      }
87
88      /* Terminate broadr. */
89      card_5
90      {
91          mat1 = 0;
92      }
93  }
94
95  unresr
96  {
97      card_1
98      {
99          nendf = -21;
100         nin = -23;
101         nout = -24;
102     }
103
104     card_2
105     {
106         matd = 1050;
107         ntemp = 3;
108         nsigz = 7;
109         iprint = 1;
110     }
111
112     card_3
113     {
114         temp[0] = 300;
115         temp[1] = 900;
116         temp[2] = 2100;
117     }
118
119     card_4
120     {
121         sigz[0] = 1.0e10;
122         sigz[1] = 1.0e5;
123         sigz[2] = 1.0e4;
124         sigz[3] = 1000.0;
125         sigz[4] = 100.0;
126         sigz[5] = 10.0;
127         sigz[6] = 1;
128     }
129
130     /* Terminate unresr. */
131     card_2
132     {
```

```
        matd = 0;
    }
}

thermr
{
    card_1
    {
        nendf = 0;
        nin = -24;
        nout = -25;
    }

    card_2
    {
        matde = 0;
        matdp = 1050;
        nbin = 8;
        ntemp = 3;
        iinc = 1;
        icoh = 0;
        natom = 1;
        mtref = 221;
        iprint = 0;
    }

    card_3
    {
        tempr[0] = 300.0; // Use C-style floats.
        tempr[1] = 900.0;
        tempr[2] = 2100.0;
    }

    card_4
    {
        tol = 0.05; // Use C-style floats.
        emax = 4.2;
    }
}

groupr
{
    card_1
    {
        nendf = -21;
        npend = -25;
        ngout1 = 0;
        ngout2 = -26;
    }

    card_2
    {
        matb = 1050;
        ign = 9;
        igg = 0;
        iwt = 5;
        lord = 3;
        ntemp = 3;
        nsigz = 7;
        iprint = 1;
    }

    card_3
    {
        title = "94-pu-238";
    }

    card_4
```

```
201     {
202         /* ntemp in card_2 denotes the number of expected temperatures. */
203         temp[0] = 300.0;
204         temp[1] = 900.0;
205         temp[2] = 2100.0;
206     }
207 
208     card_5
209     {
210         /* nsigz in card_2 denotes the number of expected sigma zeroes. */
211         sigz[0] = 1.0e10;
212         sigz[1] = 1.0e5;
213         sigz[2] = 1.0e4;
214         sigz[3] = 1000.0;
215         sigz[4] = 100.0;
216         sigz[5] = 10.0;
217         sigz[6] = 1;
218     }
219 
220     /* Reactions for temperature 300.0. */
221     card_9
222     {
223         mfd = 3;
224         mtd = 1;
225         mtname = "total";
226     }
227 
228     card_9
229     {
230         mfd = 3;
231         mtd = 2;
232         mtname = "elastic";
233     }
234 
235     card_9
236     {
237         mfd = 3;
238         mtd = 16;
239         mtname = "n2n";
240     }
241 
242     card_9
243     {
244         mfd = 3;
245         mtd = 17;
246         mtname = "n3n";
247     }
248 
249     card_9
250     {
251         mfd = 3;
252         mtd = 18;
253         mtname = "fission";
254     }
255 
256     card_9
257     {
258         mfd = 3;
259         mtd = 102;
260         mtname = "capture";
261     }
262 
263     card_9
264     {
265         mfd = 3;
266         mtd = 221;
267         mtname = "free gas thermal";
268     }
```

```
269
270       card_9
271       {
272           mfd = 6;
273           mtd = 2;
274           mtname = "elastic";
275       }
276
277       card_9
278       {
279           mfd = 6;
280           mtd = 16;
281           mtname = "n2n";
282       }
283
284       card_9
285       {
286           mfd = 6;
287           mtd = 17;
288           mtname = "n,3n";
289       }
290
291       card_9
292       {
293           mfd = 6;
294           mtd = 18;
295           mtname = "fission";
296       }
297
298       card_9
299       {
300           mfd = 6;
301           mtd = 51;
302           mtname = "discrete inelastic";
303       }
304
305       card_9
306       {
307           mfd = 6;
308           mtd = -59;
309           mtname = "continued";
310       }
311
312       card_9
313       {
314           mfd = 6;
315           mtd = 91;
316           mtname = "continuum inelastic";
317       }
318
319       card_9
320       {
321           mfd = 6;
322           mtd = 221;
323           mtname = "free gas thermal";
324       }
325
326       /* Terminate temperature 300.0. */
327       card_9
328       {
329           mfd = 0;
330       }
331
332       /* Reactions for temperature 900.0. */
333       card_9
334       {
335           mfd = 3;
336           mtd = 1;
```

```
337         mtname = "total";
338     }
339
340     card_9
341     {
342         mfd = 3;
343         mtd = 2;
344         mtname = "elastic";
345     }
346
347     card_9
348     {
349         mfd = 3;
350         mtd = 18;
351         mtname = "fission";
352     }
353
354     card_9
355     {
356         mfd = 3;
357         mtd = 102;
358         mtname = "capture";
359     }
360
361     card_9
362     {
363         mfd = 3;
364         mtd = 221;
365         mtname = "free gas thermal";
366     }
367
368     card_9
369     {
370         mfd = 6;
371         mtd = 2;
372         mtname = "elastic";
373     }
374
375     card_9
376     {
377         mfd = 6;
378         mtd = 221;
379         mtname = "free gas thermal";
380     }
381
382     /* Terminate temperature 900.0. */
383     card_9
384     {
385         mfd = 0;
386     }
387
388     /* Reactions for temperature 2100.0. */
389     card_9
390     {
391         mfd = 3;
392         mtd = 1;
393         mtname = "total";
394     }
395
396     card_9
397     {
398         mfd = 3;
399         mtd = 2;
400         mtname = "elastic";
401     }
402
403     card_9
404     {
```

```
405          mfd = 3;
406          mtd = 18;
407          mtname = "fission";
408      }
409
410      card_9
411      {
412          mfd = 3;
413          mtd = 102;
414          mtname = "capture";
415      }
416
417      card_9
418      {
419          mfd = 3;
420          mtd = 221;
421          mtname = "free gas thermal";
422      }
423
424      card_9
425      {
426          mfd = 6;
427          mtd = 2;
428          mtname = "elastic";
429      }
430
431      card_9
432      {
433          mfd = 6;
434          mtd = 221;
435          mtname = "free gas thermal";
436      }
437
438      /* Terminate temperature 2100.0. */
439      card_9
440      {
441          mfd = 0;
442      }
443
444      /* Terminate groupr. */
445      card_10
446      {
447          matd = 0;
448      }
449  }
450
451  wimsr
452  {
453      card_1
454      {
455          ngendf = -26;
456          nout = 27;
457      }
458
459      card_2
460      {
461          iprint = 1;
462      }
463
464      card_3
465      {
466          mat = 1050;
467          nfid = 1;
468          rdfid = 1050.0;
469      }
470
471      card_4
472      {
```

```
473           ntemp = 3;
474           nsigz = 7;
475           sgref = 1e10;
476           ires = 3;
477           sigp = 10.890;
478           mti = 221;
479           mtc = 0;
480       }
481
482       card_7
483       {
484           lambda[0] = 1.0;
485           lambda[1] = 1.0;
486           lambda[2] = 1.0;
487           lambda[3] = 1.0;
488           lambda[4] = 1.0;
489           lambda[5] = 1.0;
490           lambda[6] = 1.0;
491           lambda[7] = 1.0;
492           lambda[8] = 1.0;
493           lambda[9] = 1.0;
494           lambda[10] = 1.0;
495           lambda[11] = 1.0;
496           lambda[12] = 1.0;
497       }
498   }
```

Expected NJOY Input Instructions for Test Problem 11

```
 1  moder
 2  20 -21/ ### card_1
 3  reconr
 4  -21 -22/ ### card_1
 5  'pendf tape for pu-238 from endf/b-iv tape 404'/ ### card_2
 6  1050 3/ ### card_3
 7  0.005/ ### card_4
 8  '94-pu-238 from endf/b tape t404'/ ### card_5
 9  'processed by the njoy nuclear data processing system'/ ### card_5
10  'see original endf/b-iv tape for details of evaluation'/ ### card_5
11  0/ ### card_3
12  broadr
13  -21 -22 -23/ ### card_1
14  1050 3 0 1 0/ ### card_2
15  0.005/ ### card_3
16  300.0 900.0 2100.0/ ### card_4
17  0/ ### card_5
18  unresr
19  -21 -23 -24/ ### card_1
20  1050 3 7 1/ ### card_2
21  300 900 2100/ ### card_3
22  1.0e10 1.0e5 1.0e4 1000.0 100.0 10.0 1/ ### card_4
23  0/ ### card_2
24  thermr
25  0 -24 -25/ ### card_1
26  0 1050 8 3 1 0 1 221 0/ ### card_2
27  300.0 900.0 2100.0/ ### card_3
28  0.05 4.2/ ### card_4
29  groupr
30  -21 -25 0 -26/ ### card_1
31  1050 9 0 5 3 3 7 1/ ### card_2
32  '94-pu-238'/ ### card_3
33  300.0 900.0 2100.0/ ### card_4
34  1.0e10 1.0e5 1.0e4 1000.0 100.0 10.0 1/ ### card_5
35  3 1 'total'/ ### card_9
36  3 2 'elastic'/ ### card_9
37  3 16 'n2n'/ ### card_9
38  3 17 'n3n'/ ### card_9
```

```
3 18 'fission'/ ### card_9
3 102 'capture'/ ### card_9
3 221 'free gas thermal'/ ### card_9
6 2 'elastic'/ ### card_9
6 16 'n2n'/ ### card_9
6 17 'n,3n'/ ### card_9
6 18 'fission'/ ### card_9
6 51 'discrete inelastic'/ ### card_9
6 -59 'continued'/ ### card_9
6 91 'continuum inelastic'/ ### card_9
6 221 'free gas thermal'/ ### card_9
0/ ### card_9
3 1 'total'/ ### card_9
3 2 'elastic'/ ### card_9
3 18 'fission'/ ### card_9
3 102 'capture'/ ### card_9
3 221 'free gas thermal'/ ### card_9
6 2 'elastic'/ ### card_9
6 221 'free gas thermal'/ ### card_9
0/ ### card_9
3 1 'total'/ ### card_9
3 2 'elastic'/ ### card_9
3 18 'fission'/ ### card_9
3 102 'capture'/ ### card_9
3 221 'free gas thermal'/ ### card_9
6 2 'elastic'/ ### card_9
6 221 'free gas thermal'/ ### card_9
0/ ### card_9
0/ ### card_10
wimsr
-26 27/ ### card_1
1/ ### card_2
1050 1 1050.0/ ### card_3
3 7 1e10 3 10.890 221 0/ ### card_4
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0/ ### card_7
stop
```

B.11 Test Problem 12 (tp12)

NIF Version of Test Problem 12

```
reconr
{
    card_1
    {
        nendf = 20;
        npend = 21;
    }

    card_2
    {
        tlabel = "pendf tape for endf/b-vi.1 28-ni-61a";
    }

    card_3
    {
        mat = 2834;
        ncards = 1;
        ngrid = 0;
    }

    card_4
    {
        /* Note C-style float compared to the original declaration above. */
        err = 0.01;
    }

    card_5
    {
        cards = "28-ni-61a from endf/b-vi.1 t124 (hetrick,fu;ornl)";
    }

    /* Terminate execution of reconr with mat = 0 as usual. */
    card_3
    {
        mat = 0;
    }
}

gaspr
{
    card_1
    {
        nendf = 20;
        nin = 21;
        nout = 22;
    }
}

plotr
{
    card_0
    {
        nplt = 23;
    }

    card_1
    {
        lori = 1;
        istyle = 1;
        size = 0.3;
        ipcol = 2;
    }

    /* New axes, new page. */
```

```
65      card_2
66      {
67          iplot = 1;
68          iwcol = 3;
69      }
70
71      card_3
72      {
73          t1 = "<endf/b-vi n>i-61";
74      }
75      card_3a
76      {
77          t2 = "<r>esonance <c>ross <s>ections";
78      }
79
80      card_4
81      {
82          itype = 2;
83          jtype = 0;
84          igrid = 3;
85          ileg = 1;
86          xtag = 23e3;
87          ytag = 5e2;
88      }
89
90      card_5
91      {
92          el = 0.5e4;
93          eh = 3e4;
94          xstep = 0.5e4;
95      }
96      card_5a {}
97
98      card_6
99      {
100         yl = 1e-3;
101         yh = 1e3;
102     }
103     card_6a {}
104
105     /* card 7 and card 7a skipped since jtype = 0. */
106
107     card_8
108     {
109         iverf = 6;
110         nin = 22;
111         matd = 2834;
112         mfd = 3;
113         mtd = 2;
114     }
115
116     /* itype is positive, resulting in 2d plot. */
117     card_9
118     {
119         icon = 0;
120         isym = 0;
121         idash = 0;
122         iccol = 3;
123         ithick = 2;
124     }
125
126     /* ileg = 1, resulting in card 10 but no card 10a. */
127     card_10
128     {
129         aleg = "elastic";
130     }
131
132     /* card 11-13 skipped since it's a 2d plot and iverf != 0. */
```

```
133
134     /* New curve; 2nd additional plot on existing axes. */
135     card_2
136     {
137         iplot = 2;
138     }
139
140     /* card 2-7 skipped since iplot = 2. */
141
142     card_8
143     {
144         iverf = 6;
145         nin = 22;
146         matd = 2834;
147         mfd = 3;
148         mtd = 102;
149     }
150
151     /* itype is positive on the current axes, resulting in 2d plot. */
152     card_9
153     {
154         icon = 0;
155         isym = 0;
156         idash = 0;
157         iccol = 1;
158         ithick = 2;
159     }
160
161     /* ileg = 1 on current axes, resulting in card 10 but no card 10a. */
162     card_10
163     {
164         aleg = "capture";
165     }
166
167     /* New axes, new page. */
168     card_2
169     {
170         iplot = 1;
171         iwcol = 7;
172     }
173
174     card_3
175     {
176         t1 = "<endf/b-vi n>i-61";
177     }
178     card_3a
179     {
180         t2 = "<g>as <p>roduction";
181     }
182
183     card_4
184     {
185         itype = 1;
186         jtype = 0;
187         igrid = 3;
188         ileg = 1;
189     }
190
191     card_5
192     {
193         el = 0;
194         eh = 2e7;
195         xstep = 5e6;
196     }
197     card_5a {}
198
199     card_6 {}
200     card_6a {}
```

```
201
202       /* card 7 and card 7a skipped since jtype = 0. */
203
204       card_8
205       {
206           iverf = 6;
207           nin = 22;
208           matd = 2834;
209           mfd = 3;
210           mtd = 203;
211           temper = 0.0;
212       }
213
214       /* itype is positive, resulting in 2d plot. */
215       card_9
216       {
217           icon = 0;
218           isym = 0;
219           idash = 0;
220           iccol = 1;
221           ithick = 2;
222       }
223
224       /* ileg = 1, resulting in card 10 but no card 10a. */
225       card_10
226       {
227           aleg = "hydrogen";
228       }
229
230       /* card 11-13 skipped since it's a 2d plot and iverf != 0. */
231
232       /* New curve; 2nd additional plot on existing axes. */
233       card_2
234       {
235           iplot = 2;
236       }
237
238       /* card 2-7 skipped since iplot = 2. */
239
240       card_8
241       {
242           iverf = 6;
243           nin = 22;
244           matd = 2834;
245           mfd = 3;
246           mtd = 207;
247           temper = 0.0;
248       }
249
250       /* itype is positive on the current axes, resulting in 2d plot. */
251       card_9
252       {
253           icon = 0;
254           isym = 0;
255           idash = 0;
256           iccol = 2;
257           ithick = 2;
258       }
259
260       /* ileg = 1 on current axes, resulting in card 10 but no card 10a. */
261       card_10
262       {
263           aleg = "helium-4";
264       }
265
266       /* Terminate plotting job. */
267       card_2
268       {
```

```
269          iplot = 99;
270      }
271  }
272
273  viewr
274  {
275      /* Documentation names the first two cards as card 1. Use card 0 to
              denote
276         the first card, just like in plotr.
277       */
278      card_0
279      {
280          infile = 23;
281          nps = 24;
282      }
283  }
```

Expected NJOY Input Instructions for Test Problem 12

```
 1  reconr
 2  20 21/ ### card_1
 3  'pendf tape for endf/b-vi.1 28-ni-61a'/ ### card_2
 4  2834 1 0/ ### card_3
 5  0.01/ ### card_4
 6  '28-ni-61a from endf/b-vi.1 t124 (hetrick,fu;ornl)'/ ### card_5
 7  0/ ### card_3
 8  gaspr
 9  20 21 22/ ### card_1
10  plotr
11  23/ ### card_0
12  1 1 0.3 2/ ### card_1
13  1 3/ ### card_2
14  '<endf/b-vi n>i-61'/ ### card_3
15  '<r>esonance <c>ross <s>ections'/ ### card_3a
16  2 0 3 1 23e3 5e2/ ### card_4
17  0.5e4 3e4 0.5e4/ ### card_5
18  / ### card_5a
19  1e-3 1e3/ ### card_6
20  / ### card_6a
21  6 22 2834 3 2/ ### card_8
22  0 0 0 3 2/ ### card_9
23  'elastic'/ ### card_10
24  2/ ### card_2
25  6 22 2834 3 102/ ### card_8
26  0 0 0 1 2/ ### card_9
27  'capture'/ ### card_10
28  1 7/ ### card_2
29  '<endf/b-vi n>i-61'/ ### card_3
30  '<g>as <p>roduction'/ ### card_3a
31  1 0 3 1/ ### card_4
32  0 2e7 5e6/ ### card_5
33  / ### card_5a
34  / ### card_6
35  / ### card_6a
36  6 22 2834 3 203 0.0/ ### card_8
37  0 0 0 1 2/ ### card_9
38  'hydrogen'/ ### card_10
39  2/ ### card_2
40  6 22 2834 3 207 0.0/ ### card_8
41  0 0 0 2 2/ ### card_9
42  'helium-4'/ ### card_10
43  99/ ### card_2
44  viewr
45  23 24/ ### card_0
46  stop
```

B.12 Test Problem 13 (tp13)

NIF Version of Test Problem 13

```
moder
{
    card_1
    {
        nin = 20;
        nout = -21;
    }
}

reconr
{
    card_1
    {
        nendf = -21;
        npend = -22;
    }

    card_2
    {
        tlabel = "pendf tape for endf/b-vi.1 28-ni-61a";
    }

    card_3
    {
        mat = 2834;
        ncards = 1;
        ngrid = 0;
    }

    card_4
    {
        err = 0.01;
    }

    card_5
    {
        cards = "28-ni-61a from endf/b-vi.1 t124 (hetrick,fu;ornl)";
    }

    card_3
    {
        mat = 0;
    }
}

broadr
{
    card_1
    {
        nendf = -21;
        nin = -22;
        nout = -23;
    }

    card_2
    {
        mat1 = 2834;
        ntemp2 = 1;
    }

    card_3
    {
        errthn = 0.01;
    }
```

```
65
66      card_4
67      {
68          temp2[0] = 300;
69      }
70
71      card_5
72      {
73          mat1 = 0;
74      }
75  }
76
77  heatr
78  {
79      card_1
80      {
81          nendf = -21;
82          nin = -23;
83          nout = -24;
84          /* nplot is not required? */
85      }
86
87      card_2
88      {
89          matd = 2834;
90          npk = 6;
91          nqa = 0;
92          ntemp = 1;
93          local = 0;
94          iprint = 2;
95      }
96
97      card_3
98      {
99          /* npk = 6 -> 6 values for mtk */
100         /* Note that mtk has been defined as an array. */
101         mtk[0] = 302;
102         mtk[1] = 303;
103         mtk[2] = 304;
104         mtk[3] = 402;
105         mtk[4] = 443;
106         mtk[5] = 444;
107     }
108 }
109
110 gaspr
111 {
112     card_1
113     {
114         nendf = -21;
115         nin = -24;
116         nout = -25;
117     }
118 }
119
120 moder
121 {
122     card_1
123     {
124         nin = -25;
125         nout = 28;
126     }
127 }
128
129 acer
130 {
131     card_1
132     {
```

```
133          nendf = -21;
134          npend = -25;
135          ngend = 0;
136          nace = 26;
137          ndir = 27;
138      }
139
140      card_2
141      {
142          iopt = 1;
143          iprint = 0;
144          ntype = 1;
145      }
146
147      card_3
148      {
149          hk = "28-ni-61a endf-vi.1 njoy99";
150      }
151
152      card_5
153      {
154          matd = 2834;
155          tempd = 300;
156      }
157
158      card_6 {}
159      card_7 {}
160  }
161
162  acer
163  {
164      card_1
165      {
166          nendf = 0;
167          npend = 26;
168          ngend = 33;
169          nace = 34;
170          ndir = 35;
171      }
172
173      card_2
174      {
175          iopt = 7;
176          iprint = 1;
177          ntype = 2;
178      }
179
180      card_3
181      {
182          hk = "28-ni-61a endf-vi.1 njoy99";
183      }
184  }
185
186  viewr
187  {
188      /* Documentation names the first two cards as card 1. Use card 0 to
             denote
189         the first card, just like in plotr.
190       */
191      card_0
192      {
193          infile = 33;
194          nps = 36;
195      }
196  }
```

Expected NJOY Input Instructions for Test Problem 13

```
moder
20 -21/ ### card_1
reconr
-21 -22/ ### card_1
'pendf tape for endf/b-vi.1 28-ni-61a'/ ### card_2
2834 1 0/ ### card_3
0.01/ ### card_4
'28-ni-61a from endf/b-vi.1 t124 (hetrick,fu;ornl)'/ ### card_5
0/ ### card_3
broadr
-21 -22 -23/ ### card_1
2834 1/ ### card_2
0.01/ ### card_3
300/ ### card_4
0/ ### card_5
heatr
-21 -23 -24/ ### card_1
2834 6 0 1 0 2/ ### card_2
302 303 304 402 443 444/ ### card_3
gaspr
-21 -24 -25/ ### card_1
moder
-25 28/ ### card_1
acer
-21 -25 0 26 27/ ### card_1
1 0 1/ ### card_2
'28-ni-61a endf-vi.1 njoy99'/ ### card_3
2834 300/ ### card_5
/ ### card_6
/ ### card_7
acer
0 26 33 34 35/ ### card_1
7 1 2/ ### card_2
'28-ni-61a endf-vi.1 njoy99'/ ### card_3
viewr
33 36/ ### card_0
stop
```

B.13 Test Problem 14 (tp14)

NIF Version of Test Problem 14

```
acer
{
    card_1
    {
        endf_input = 20;
        pendf_input = 21;
        multigroup_photon_input = 0;
        ace_output = 31;
        mcnp_directory_output = 32;
    }

    card_2
    {
        acer_run_option = 1;
        print_control = 0;
        ace_output_type = 1;

        /* id suffix for zaid (default = 0.00), and
           number of iz,aw pairs to read in (default = 0) are set to their
           default values since they are not provided.
         */
    }

    card_3
    {
        description = "proton + 7-n-14 apt la150 njoy99 mcnpx";
    }

    card_5
    {
        material = 725;
        temperature = 0; // No trailing dots allowed. Use C-style floats.
    }

    /* Card 6 and 7 are empty; the default values will be used. */
    card_6 {} // Use new cummulative angle distributions.
    card_7 {} // No thinning.
}

acer
{
    card_1
    {
        endf_input = 0;
        pendf_input = 31;
        multigroup_photon_input = 33;
        ace_output = 34;
        mcnp_directory_output = 35;
    }

    card_2
    {
        acer_run_option = 7;
        print_control = 1;
        ace_output_type = 2;
    }

    card_3
    {
        description = "proton + 7-n-14 apt la150 njoy99 mcnpx";
    }
}

viewr
```

```
65  {
66      /* Documentation names the first two cards as card 1. Use card 0 to
            denote
67         the first card, just like in plotr.
68       */
69      card_0
70      {
71          input = 33;
72          output = 36;
73      }
74  }
75
76  /* The translator appends the 'stop' instruction, no neep to explicitly
77     declare it.
78   */
```

Expected NJOY Input Instructions for Test Problem 14

```
1   acer
2   20 21 0 31 32/ ### card_1
3   1 0 1/ ### card_2
4   'proton + 7-n-14 apt la150 njoy99 mcnpx'/ ### card_3
5   725 0/ ### card_5
6   / ### card_6
7   / ### card_7
8   acer
9   0 31 33 34 35/ ### card_1
10  7 1 2/ ### card_2
11  'proton + 7-n-14 apt la150 njoy99 mcnpx'/ ### card_3
12  viewr
13  33 36/ ### card_0
14  stop
```

B.14 Test Problem 17 (tp17)

NIF Version of Test Problem 17

```
reconr
{
    card_1
    {
        nendf = 21;
        npend = 41;
    }

    card_2
    {
        tlabel = "processing jendl-3.3 238u.";
    }

    card_3
    {
        mat = 9237;
        ncards = 0;
        ngrid = 0;
    }

    card_4
    {
        err = 0.001;
    }

    card_3
    {
        mat = 0;
    }
}

broadr
{
    card_1
    {
        nendf = 21;
        nin = 41;
        nout = 31;
    }

    card_2
    {
        mat1 = 9237;
        ntemp2 = 1;
        istart = 0;
        istrap = 0;
        temp1 = 0;
    }

    card_3
    {
        errthn = 0.001;
    }

    card_4
    {
        temp2[0] = 300.0;
    }

    card_5
    {
        mat1 = 0;
    }
}
```

```
65
66  reconr
67  {
68      card_1
69      {
70          nendf = 22;
71          npend = 42;
72      }
73
74      card_2
75      {
76          tlabel = "processing jendl-3.3 235u.";
77      }
78
79      card_3
80      {
81          mat = 9228;
82          ncards = 0;
83          ngrid = 0;
84      }
85
86      card_4
87      {
88          err = 0.001;
89      }
90
91      card_3
92      {
93          mat = 0;
94      }
95  }
96
97  broadr
98  {
99      card_1
100     {
101         nendf = 22;
102         nin = 42;
103         nout = 32;
104     }
105
106     card_2
107     {
108         mat1 = 9228;
109         ntemp2 = 1;
110         istart = 0;
111         istrap = 0;
112         temp1 = 0;
113     }
114
115     card_3
116     {
117         errthn = 0.001;
118     }
119
120     card_4
121     {
122         temp2[0] = 300.0;
123     }
124
125     card_5
126     {
127         mat1 = 0;
128     }
129 }
130
131 reconr
132 {
```

```
    card_1
    {
        nendf = 23;
        npend = 43;
    }

    card_2
    {
        tlabel = "processing jendl-3.3 239pu.";
    }

    card_3
    {
        mat = 9437;
        ncards = 0;
        ngrid = 0;
    }

    card_4
    {
        err = 0.001;
    }

    card_3
    {
        mat = 0;
    }
}

broadr
{
    card_1
    {
        nendf = 23;
        nin = 43;
        nout = 33;
    }

    card_2
    {
        mat1 = 9437;
        ntemp2 = 1;
        istart = 0;
        istrap = 0;
        temp1 = 0;
    }

    card_3
    {
        errthn = 0.001;
    }

    card_4
    {
        temp2[0] = 300.0;
    }

    card_5
    {
        mat1 = 0;
    }
}

groupr
{
    card_1
    {
        nendf = 21;
```

```
201         npend = 31;
202         ngout1 = 0;
203         ngout2 = 91;
204     }
205
206     card_2
207     {
208         matb = 9237;
209         ign = 3;
210         igg = 0;
211         iwt = 6;
212         lord = 1;
213         ntemp = 1;
214         nsigz = 1;
215         iprint = 0;
216     }
217
218     card_3
219     {
220         title = "u-238";
221     }
222
223     card_4
224     {
225         temp[0] = 300.0;
226     }
227
228     card_5
229     {
230         sigz[0] = 1.0e10; // No trailing dots. Use C-style floats.
231     }
232
233     card_9
234     {
235         mfd = 3;
236         /* mtd and mtname does not have to be supplied? */
237     }
238
239     card_9
240     {
241         mfd = 3;
242         mtd = 251;
243         mtname = "mubar";
244     }
245
246     card_9
247     {
248         mfd = 3;
249         mtd = 252;
250         mtname = "xi";
251     }
252
253     card_9
254     {
255         mfd = 3;
256         mtd = 452;
257         mtname = "nu";
258     }
259
260     card_9
261     {
262         mfd = 3;
263         mtd = 455;
264         mtname = "nu";
265     }
266
267     card_9
268     {
```

```
269          mfd = 3;
270          mtd = 456;
271          mtname = "nu";
272      }
273
274      card_9
275      {
276          mfd = 5;
277          mtd = 18;
278          mtname = "xi";
279      }
280
281      /* Terminate temperature/material with mfd = 0 as usual. */
282      card_9
283      {
284          mfd = 0;
285      }
286
287      /* Terminate groupr run with matd = 0 as usual. */
288      card_10
289      {
290          matd = 0;
291      }
292  }
293
294  groupr
295  {
296      card_1
297      {
298          nendf = 22;
299          npend = 32;
300          ngout1 = 0;
301          ngout2 = 92;
302      }
303
304      card_2
305      {
306          matb = 9228;
307          ign = 3;
308          igg = 0;
309          iwt = 6;
310          lord = 1;
311          ntemp = 1;
312          nsigz = 1;
313          iprint = 0;
314      }
315
316      card_3
317      {
318          title = "u-235";
319      }
320
321      card_4
322      {
323          temp[0] = 300.0;
324      }
325
326      card_5
327      {
328          sigz[0] = 1.0e10; // No trailing dots. Use C-style floats.
329      }
330
331      card_9
332      {
333          mfd = 3;
334          /* mtd and mtname does not have to be supplied? */
335      }
336
```

```
337      /* Terminate temperature/material with mfd = 0 as usual. */
338      card_9
339      {
340          mfd = 0;
341      }
342
343      /* Terminate groupr run with matd = 0 as usual. */
344      card_10
345      {
346          matd = 0;
347      }
348  }
349
350  groupr
351  {
352      card_1
353      {
354          nendf = 23;
355          npend = 33;
356          ngout1 = 0;
357          ngout2 = 93;
358      }
359
360      card_2
361      {
362          matb = 9437;
363          ign = 3;
364          igg = 0;
365          iwt = 6;
366          lord = 1;
367          ntemp = 1;
368          nsigz = 1;
369          iprint = 0;
370      }
371
372      card_3
373      {
374          title = "pu-239";
375      }
376
377      card_4
378      {
379          temp[0] = 300.0;
380      }
381
382      card_5
383      {
384          sigz[0] = 1.0e10; // No trailing dots. Use C-style floats.
385      }
386
387      card_9
388      {
389          mfd = 3;
390          /* mtd and mtname does not have to be supplied? */
391      }
392
393      /* Terminate temperature/material with mfd = 0 as usual. */
394      card_9
395      {
396          mfd = 0;
397      }
398
399      /* Terminate groupr run with matd = 0 as usual. */
400      card_10
401      {
402          matd = 0;
403      }
404  }
```

```
405
406 moder
407 {
408     card_1
409     {
410         nin = 2;
411         nout = 99;
412     }
413
414     card_2
415     {
416         tpid = "merge u235, u-238 and pu-239";
417     }
418
419     card_3
420     {
421         nin = 92;
422         matd = 9228;
423     }
424
425     card_3
426     {
427         nin = 91;
428         matd = 9237;
429     }
430
431     card_3
432     {
433         nin = 93;
434         matd = 9437;
435     }
436
437     /* Terminate moder by setting nin = 0. */
438     card_3
439     {
440         nin = 0;
441     }
442 }
443
444 errorr
445 {
446     card_1
447     {
448         nendf = 21;
449         npend = 0;
450         ngout = 99;
451         nout = 26;
452         nin = 0;
453         nstan = 0;
454     }
455
456     card_2
457     {
458         matd = 9237;
459         ign = 3;
460         iwt = 6;
461         iprint = 1;
462     }
463
464     /* Test problem 17 is using a file of the endf-5 format (iverf = 5) */
465
466     card_7
467     {
468         iread = 2;
469         mfcov = 33;
470         irespr = 1;
471         legord = 1;
472         ifissp = -1;
```

```
473     }
474
475     card_10
476     {
477         mat1 = 9228;
478         mt1 = 18;
479     }
480
481     card_10
482     {
483         mat1 = 9437;
484         mt1 = 18;
485     }
486
487     card_10
488     {
489         mat1 = 0;
490     }
491 }
```

Expected NJOY Input Instructions for Test Problem 17

```
 1  reconr
 2  21 41/ ### card_1
 3  'processing jendl-3.3 238u.'/ ### card_2
 4  9237 0 0/ ### card_3
 5  0.001/ ### card_4
 6  0/ ### card_3
 7  broadr
 8  21 41 31/ ### card_1
 9  9237 1 0 0 0/ ### card_2
10  0.001/ ### card_3
11  300.0/ ### card_4
12  0/ ### card_5
13  reconr
14  22 42/ ### card_1
15  'processing jendl-3.3 235u.'/ ### card_2
16  9228 0 0/ ### card_3
17  0.001/ ### card_4
18  0/ ### card_3
19  broadr
20  22 42 32/ ### card_1
21  9228 1 0 0 0/ ### card_2
22  0.001/ ### card_3
23  300.0/ ### card_4
24  0/ ### card_5
25  reconr
26  23 43/ ### card_1
27  'processing jendl-3.3 239pu.'/ ### card_2
28  9437 0 0/ ### card_3
29  0.001/ ### card_4
30  0/ ### card_3
31  broadr
32  23 43 33/ ### card_1
33  9437 1 0 0 0/ ### card_2
34  0.001/ ### card_3
35  300.0/ ### card_4
36  0/ ### card_5
37  groupr
38  21 31 0 91/ ### card_1
39  9237 3 0 6 1 1 1 0/ ### card_2
40  'u-238'/ ### card_3
41  300.0/ ### card_4
42  1.0e10/ ### card_5
43  3/ ### card_9
44  3 251 'mubar'/ ### card_9
45  3 252 'xi'/ ### card_9
```

```
3 452 'nu'/ ### card_9
3 455 'nu'/ ### card_9
3 456 'nu'/ ### card_9
5 18 'xi'/ ### card_9
0/ ### card_9
0/ ### card_10
groupr
22 32 0 92/ ### card_1
9228 3 0 6 1 1 1 0/ ### card_2
'u-235'/ ### card_3
300.0/ ### card_4
1.0e10/ ### card_5
3/ ### card_9
0/ ### card_9
0/ ### card_10
groupr
23 33 0 93/ ### card_1
9437 3 0 6 1 1 1 0/ ### card_2
'pu-239'/ ### card_3
300.0/ ### card_4
1.0e10/ ### card_5
3/ ### card_9
0/ ### card_9
0/ ### card_10
moder
2 99/ ### card_1
'merge u235, u-238 and pu-239'/ ### card_2
92 9228/ ### card_3
91 9237/ ### card_3
93 9437/ ### card_3
0/ ### card_3
errorr
21 0 99 26 0 0/ ### card_1
9237 3 6 1/ ### card_2
2 33 1 1 -1/ ### card_7
9228 18/ ### card_10
9437 18/ ### card_10
0/ ### card_10
stop
```