

JAERI-M
89-184

原子炉異常診断エキスパートシステム
DISKET

1989年11月

吉田 一雄・横林 正雄

JAERI-Mレポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の問い合わせは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）
あて、お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11茨城
県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division, Department
of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun,
Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1989

編集兼発行 日本原子力研究所
印刷 日立高速印刷株式会社

原子炉異常診断エキスパートシステム

D I S K E T

日本原子力研究所東海研究所原子炉安全工学部

吉田 一雄・横林 正雄

(1989年10月16日受理)

原子炉異常診断システムDISKETは、原子力発電所において異常が発生した場合に、その種類および原因の診断を行うシステムとして、原研において開発したシステムである。本システムは、原因同定の手法として人工知能の一分野である知識工学を応用しており、診断のための推論を行う推論エンジン「IERIAS」と、それに必要な知識を納めた知識ベースから成る。また、本システムの特徴として、事象の時間的変化を推論で考慮できること、大量のルールを効率的に処理するために知識ベースがユニット化されていること等が挙げられる。

DISKETは、原子炉の異常診断を目指して開発されたが、推論エンジンと知識ベースが完全に分離独立しているので、推論に必要な知識がIF-THEN形式のルールで表現できる分野では、DISKETを利用して診断、推論が行える。

本報告書は、DISKETの利用者の便に供するためにまとめたものであり、システム全体の概説、知識ベース作成手引、およびシステム操作の手引から成る。

Reactor Accident Diagnostic Expert System
DISKET

Kazuo YOSHIDA and Masao YOKOBAYASHI

Department of Reactor Safety Research
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received October 16, 1989)

A reactor accident diagnostic system DISKET has been developed to identify the cause and the type of an abnormal transient of a nuclear power plant. The system is based on the knowledge engineering and consists of an inference engine IERIAS and a knowledge base. The main features of DISKET are the following: Time-varying characteristics of transient can be treated and knowledge base can be divided into several knowledge units to handle a lot of rules effectively.

This report has been provided for the convenience of DISKET's users and consists of three parts. The first part is the description of the whole system, the details of the knowledge base of DISKET are described in the second part, and how to use the DISKET system is explained in the third part.

Keywords: DISKET Code, Reactor Accidents, Diagnostic System, Expert Systems, Users Manual

目 次

1. DISKETシステムの概要	1
1.1 推論機構 I E R I A S	2
1.1.1 知識ベース開発支援機能	2
1.1.2 知識ベースコンパイル機能	3
1.1.3 ケースデータ作成・編集機能	4
1.1.4 推論機能	4
1.1.5 その他の機能	4
1.2 知識ベースの構造	5
1.2.1 データ定義部	5
1.2.2 ルール部	6
1.2.3 実行処理定義部	7
1.2.4 ルール表現における時間の取扱い	8
1.3 推論	9
1.3.1 包括型推論の手順	9
1.3.2 確信度	10
2. 知識ベース作成手引き	34
2.1 知識表現	34
2.1.1 知識ベースの構造	34
2.1.2 知識ベースの記述形式	34
2.1.3 使用文字の制約	34
2.2 データ定義部	37
2.2.1 仮説定義	37
2.2.2 生起事象定義	41
2.3 ルール部	47
2.3.1 知識ユニット定義	48
2.3.2 ルールの定義	51
2.4 実行処理定義部	56
2.4.1 アクション定義	56
3. 操作マニュアル	59
3.1 操作の概要	59
3.2 モードの選択	61
3.2.1 起動	61
3.2.2 モード選択	61
3.2.3 終了	63

3.3	モードの操作	64
3.3.1	全機能モード	65
3.3.2	知識ベースの構築支援モード	66
3.3.3	コンパイルモード	66
3.3.4	RUNモード	67
3.3.5	INFERモード	67
3.4	各機能の操作	68
3.4.1	知識ベースの構築支援	68
3.4.2	コンパイル	81
3.4.3	ケースジェネレータ	82
3.4.4	エディタ	84
3.4.5	ロード	93
3.4.6	セーブ	94
3.4.7	推論実行	95
3.4.8	コンサルテーション	99
3.4.9	計算コードの制御	113
3.4.10	知識ベースの検索	116
3.4.11	知識ベースのテスト	122
3.5	操作上の注意	125
3.6	エラーメッセージ一覧	126
	参考文献	130
	付録A UTILISPの概要	131
	付録B DISKETのファイル構成	135
	付録C 知識ベースの例	139

Contents

1. Overview of DISKET System	1
1.1 Inference Engine IERIAS	2
1.1.1 Function of Support for Knowledge Base Building	2
1.1.2 Function of Compiling Knowledge Base	3
1.1.3 Function of Generating and Editing of Case Data	4
1.1.4 Function of Inference	4
1.1.5 Miscellaneous Functions	4
1.2 Structure of Knowledge Base	5
1.2.1 Data Definition Part	5
1.2.2 Rule Part.....	6
1.2.3 Action Definition Part	7
1.2.4 Consideration of Time Histories in Rule Representation	8
1.3 Inference	9
1.3.1 Procedure of Comprehensive Inference	9
1.3.2 Certainty Factor	10
2. Instruction of Knowledge Base Building	34
2.1 Knowledge Representation	34
2.1.1 Structure of Knowledge Base	34
2.1.2 Format of Knowledge Base Representation.....	34
2.1.3 Restriction of Letter in Knowledge Base	34
2.2 Data Definition Part	37
2.2.1 Definition of Hypotheses	37
2.2.2 Definition of Findings	41
2.3 Rule Part	47
2.3.1 Definition of Knowledge Units	48
2.3.2 Definition of Rules	51
2.4 Action Definition Part	56
2.4.1 Definition of Actions	56
3. User's Instruction	59
3.1 Overview of Mode Configuration	59
3.2 Mode Selection	61
3.2.1 System Open	61
3.2.2 Mode Selection	61
3.2.3 System Close	63

3.3	Function of Each Mode	64
3.3.1	All Function Mode	65
3.3.2	Knowledge Base Building Mode	66
3.3.3	Compile Mode	66
3.3.4	RUN Mode	67
3.3.5	INFER Mode	67
3.4	Comand Description	68
3.4.1	Support for Knowledge Base Building	68
3.4.2	Compiling	81
3.4.3	Generating of Case File	82
3.4.4	Editting of Case File and Knowledge Base	84
3.4.5	Loading of Case File and Knowledge Base	93
3.4.6	Saving of Case File and Knowledge Base	94
3.4.7	Inference	95
3.4.8	Consultation	99
3.4.9	Setup of Transient Simulation	113
3.4.10	Retrieval of Knowledge Base	116
3.4.11	Test of Knowledge Base	122
3.5	Miscellaneous Information	125
3.6	Error Messages	126
Reference	130
Appendix A	Overview of UTILISP	131
Appendix B	File Configuration of DISKET	135
Appendix C	Example of Knowledge Base	139

表 一 覧

表 1. 1	予測コンサルテーションメニュー
表 2. 1	文字の使用制限
表 3. 1	コマンドプロシジャー; D I S K E Tのparameter
表 3. 2	コマンドプロシジャー; D I S K E Tのsub-parameter
表 3. 3	全機能モードでのメニューと下位機能
表 3. 4	知識ベース構築支援メニュー一覧
表 3. 5	知識ベースエディタコマンド
表 3. 6	行コマンド
表 3. 7	エディタコマンド
表 3. 8	データ定義編集コマンド
表 3. 9	データ定義挿入コマンドオプション
表 3. 10	データ定義変更コマンドオプション
表 3. 11	知識ユニット編集コマンド
表 3. 12	ケースデータ編集コマンド (その1)
表 3. 13	ケースデータ編集コマンド (その2)
表 3. 14	推論実行時の間合わせ入力一覧
表 3. 15	コンサルテーションメニュー一覧
表 3. 16	L I S Tコマンド
表 A. 1	磁気テープの内容とD A S D上での大きさ
表 A. 2	UTILISPの保存ファイル
表 B. 1	D I S K E Tのファイル構成
表 B. 2	D I S K E Tで使用するファイル
表 B. 3	メンバー一覧

図 一 覧

- 図 1. 1 DISKETシステムの概念図
 - 図 1. 2 決定木の出力例
 - 図 1. 3 ルール説明機能の出力例
 - 図 1. 4 画面二分割
 - 図 1. 5 ケースデータの構造
 - 図 1. 6 ケースデータの例
 - 図 1. 7 トレース実行例
 - 図 1. 8 知識ベースの構造
 - 図 1. 9 知識ベース内の仮説ブロック
 - 図 1. 10 コンパイル後の仮説分類の構造
 - 図 1. 11 コンパイル後の対応措置の構造
 - 図 1. 12 コンパイル後の内部仮説の構造
 - 図 1. 13 コンパイル後の仮説の構造
 - 図 1. 14 知識ベース内の生起事象ブロック
 - 図 1. 15 生起事象の構造
 - 図 1. 16 コンパイルされた生起事象
 - 図 1. 17 DISKETシステムのルール例
 - 図 1. 18 コンパイル後のDISKETシステムのルール表現
 - 図 1. 19 知識ユニットの例
 - 図 1. 20 コンパイル後の知識ユニット
 - 図 1. 21 実行処理定義の例
 - 図 1. 22 推論の流れ
 - 図 1. 23 包括型推論におけるルールの適用手順
-
- 図 2. 1 知識ベースの記述形式
 - 図 2. 2 データ定義部の形式
 - 図 2. 3 階層的仮説（仮説分類）定義例
 - 図 2. 4 仮説分類定義形式
 - 図 2. 5 仮説分類定義例
 - 図 2. 6 内部仮説定義形式
 - 図 2. 7 内部仮説定義例
 - 図 2. 8 対応措置定義形式
 - 図 2. 9 対応措置定義例
 - 図 2. 10 Rタイプの生起事象のルール中での使用形式
 - 図 2. 11 生起事象値リスト
 - 図 2. 12 生起事象の定義形式

- 図 2. 1 3 一値選択型事象定義形式
- 図 2. 1 4 一値選択型事象定義例
- 図 2. 1 5 多岐選択型事象定義形式
- 図 2. 1 6 多岐選択型事象定義例
- 図 2. 1 7 YES/NO型事象定義形式
- 図 2. 1 8 YES/NO型事象定義例
- 図 2. 1 9 数値型事象定義形式
- 図 2. 2 0 数値型事象定義例
- 図 2. 2 1 ルール部定義形式
- 図 2. 2 2 知識ユニット定義形式
- 図 2. 2 3 知識ユニットの適用条件定義例
- 図 2. 2 4 FFルールの定義例
- 図 2. 2 5 FHルールの定義例
- 図 2. 2 6 HHルールの定義例
- 図 2. 2 7 実行処理定義形式
- 図 2. 2 8 アクション定義形式
- 図 2. 2 9 アクション定義例

- 図 3. 1 DISKETの操作の概要
- 図 3. 2 DISKETの起動と終了
- 図 3. 3 初期画面 (全機能)
- 図 3. 4 知識ベース構築支援モード初期画面
- 図 3. 5 コンパイルモード
- 図 3. 6 RUNモード
- 図 3. 7 INFERRモード
- 図 3. 8 知識ベース構築支援初期画面
- 図 3. 9 知識ベース構築支援画面
- 図 3. 1 0 知識ベース・エディタ画面
- 図 3. 1 1 決定木出力
- 図 3. 1 2 決定木出力の入力例
- 図 3. 1 3 生起事象値の設定
- 図 3. 1 4 知識ベース表示画面
- 図 3. 1 5 画面二分割
- 図 3. 1 6 関連ルール表示機能の実行例
- 図 3. 1 7 ルール説明機能の実行例
- 図 3. 1 8 関連事象表示機能の実行例
- 図 3. 1 9 コンパイル実行例
- 図 3. 2 0 checklistのケースデータ作成

- 図3. 2 1 multiple choiceのケースデータ作成
- 図3. 2 2 numericalのケースデータ作成
- 図3. 2 3 YES/NOのケースデータ作成
- 図3. 2 4 ケースジェネレータ実行例
- 図3. 2 5 エディタの初期表示
- 図3. 2 6 ケースデータ編集実行例
- 図3. 2 7 ロード実行例
- 図3. 2 8 セーブ実行例
- 図3. 2 9 推論実行時の質問形式
- 図3. 3 0 推論結果
- 図3. 3 1 推論結果保存の問合わせ
- 図3. 3 2 出力コントロール用パラメータの変更
- 図3. 3 3 出力コントロール用パラメータ変更後の推論結果出力
- 図3. 3 4 不明生起事象の出力
- 図3. 3 5 否定ルールの出力
- 図3. 3 6 排反ルールの出力
- 図3. 3 7 生起事象値の出力
- 図3. 3 8 生起事象値の設定
- 図3. 3 9 推論過程の表示
- 図3. 4 0 全生起事象の出力
- 図3. 4 1 排反生起事象の出力
- 図3. 4 2 ルールの内容説明
- 図3. 4 3 ルールの部分修正
- 図3. 4 4 RECONFの出力メッセージ
- 図3. 4 5 指定生起事象の再確認
- 図3. 4 6 全生起事象の再確認
- 図3. 4 7 DETAILの出力メッセージ
- 図3. 4 8 詳細事故診断の実行結果(その1)
- 図3. 4 9 詳細事故診断の実行結果(その2)
- 図3. 5 0 詳細事故診断の実行結果(その3)
- 図3. 5 1 詳細事故診断の実行結果(その4)
- 図3. 5 2 シミュレーションプログラムの起動
- 図3. 5 3 指定生起事象を含むルールの出力
- 図3. 5 4 指定生起事象を条件部に含むルールの出力
- 図3. 5 5 システムステータスの表示
- 図3. 5 6 計算コードを利用するときの知識ベースの表現方法
- 図3. 5 7 仮説の検索の実行例
- 図3. 5 8 仮説分類(事故仮説)の検索の実行例

- 図 3. 5 9 中間仮説の検索の実行例
- 図 3. 6 0 対応措置（対応仮説）の検索の実行例
- 図 3. 6 1 生起事象の検索の実行例
- 図 3. 6 2 知識ベース表示の実行例
- 図 3. 6 3 知識ユニットの検索の実行例
- 図 3. 6 4 ルールの検索の実行例
- 図 3. 6 5 ルールのトレース
- 図 3. 6 6 ルールのバックトレース

1. DISKETシステムの概要

TMI (Three Mile Island Unit 2) の事故以後、原子力発電所の異常時に備えて、事故の未然防止および影響の軽減を目的とした計算機を利用した運転支援システムの開発、提案が、数多く成されている。DISKETシステムも上記の目的のうち特に事故原因の同定を目的として開発されたシステムである。

本システムの特徴は、原因同定の手法として人工知能の一分野である知識工学を応用している点である。知識工学的手法の原子力分野への応用を目標とした研究は近年、盛んに行われるようになってきたが、元来、同手法は医療診断等の静的事象の分野で発展した技術である。

原研においても、本システムの開発に先立って、医療診断を目的として開発された EXPERTシステム¹⁾を用いて、同手法の原子炉異常診断への利用可能性の検討²⁾を行った。その結果、事象の時間的変化の推論への考慮、大量のルールの高速かつ効率的処理が重要であることが明らかになった。この検討結果を踏まえ、原研で独自に開発したエキスパートシステムがDISKETシステム³⁾⁴⁾である。

図1. 1に示すようにDISKETシステムは、通常のエキスパートシステムと同様に推論エンジン IERIAS⁵⁾と知識ベースから成る。推論エンジンは、知識ベースと、異常状態にある原子炉からのプロセスデータを基に、異常の種類、原因の同定を行う。図中の黒の矢印が推論実行時のデータの流れを示す。知識ベースは推論に必要な情報（ルールと呼ぶ）を蓄えたものであり、その知識獲得の経路は図中に白矢印で示されている。すなわち、原子力プラントの各コンポーネント、ポンプ、バルブ等の動作条件、幾何形状などの設計データ、あるいは過去の運転記録または、熱水力計算コードの解析結果等から得られる過渡変化時のプラントの動特性データ等が、ルール表現に変換されて知識ベースに格納される。

1. 1 推論機構 I E R I A S

推論機構 I E R I A S は、原研において新たに開発したプログラムである。使用言語は U T I L I S P (1) であり、原研計算センターの F A C O M M 7 8 0 上で稼働している。I E R I A S の特徴として次項を挙げることができる。

- (a) 圧力、温度等のプラントのパラメータの時間変化、警報の発信時刻、弁、ポンプの作動時刻等、過渡変化の特徴を推論過程で考慮できるようになった。
- (b) プログラミング言語として記号処理に適した L i s p の一方言の U T I L I S P を使用したことで、ルール表現の自由度が広がった。
- (c) オンライン診断あるいは速いデータ処理を可能とするため、データの一括入力ができる。
- (d) 推論の効率化を計るため知識ベースの分割（ユニット化）制御を可能にした。
- (e) 確信度付きの推論が可能のため柔軟性のある診断が行なえる。
- (f) 事故の予測や対応策の予測のため計算コードを自動起動することが可能。

I E R I A S は、推論を実行するだけでなく、推論に不可欠な知識ベースの作成編集を容易にするための”知識ベース開発支援機能”等が設けられている。以下に I E R I A S の有する各機能の概略を記す。

1. 1. 1 知識ベース開発支援機能

知識ベースは、本システムの場合、専門家の知識を I F - T H E N 形式のプロダクションルールで表現したものの集合であり、診断対象ごとに作成する必要がある。知識ベース開発支援機能は、知識ベース作成を容易にするために整備された機能である。本機能は、大別すると、知識ベースを作成・編集するための機能と、作成済の知識ベースの構造、あるいはルール間の関係等を解釈しその結果を T S S 画面上に表示するなどして知識ベースの理解、内容検討の手助けを行うための機能に分れる。以下にその概略を記す。

i) 知識ベース編集機能

知識ベースエディタは、ファイルに保存されている知識ベースを、T S S 画面上に表示し、一画面上で文字単位、行単位の追加、変更、削除の編集が可能である（詳細は 3.4.1 節の vi) を参照のこと）。

ii) 決定木出力機能

決定木出力は、入力で指定した仮説について関連するルールや条件を Tree 状に表示する

機能である。その例を図 1. 2 に示す（詳細は 3.4.1 節の iv) を参照のこと）。

iii) ルール説明機能

D I S K E T の知識ベース内でのルールは、図 1. 3 上段に示すような表現形式になっており、生起事象や仮説が、英数字の略称名で表わされている。この表記法はルールの内容を一目見て理解することを困難にしており、これを改善するために整備されたのがルール説明機能である。この機能を利用すると、図 1. 3 下段に示すように、ルールがより自然言語に近い表現になって表示される（詳細は 3.4.1 節の viii) を参照のこと）。

iv) 確信度簡易計算機能

本機能は、決定木出力機能と併用することによって、特定の仮説の確信度を計算する機能である。確信度計算に必要な生起事象の値は、ケースファイル（推論のためのアラーム等のデータを一まとめにしたファイル）からの一括入力あるいは、個々に入力することも可能である。生起事象の値を一度設定した後は、個々の生起事象の値を変えることにより、確信度への影響を調べることができる（詳細は 3.4.1 節の v) を参照のこと）。

v) 画面二分割機能

C R T 画面を上下に二分割して、それぞれに異なった内容を表示する機能であり、前述の i) ~ iv) の機能のうち、どれか二つの表示内容を同時に一画面に表示できる。たとえば、上部画面で、決定木を出力し、下部画面では、決定木中に含まれるルールについてルール説明機能を用いて、その自然語表示を行うというような利用が可能である。その一例を図 1. 4 に示す。

1. 1. 2 知識ベースコンパイル機能

前述の知識ベース開発支援機能を用いて作成した知識ベースの知識表現は、利用者による知識ベースの作成、保守が容易な形式になっており、逆に推論実行の面から見ると効率のよい形式ではない。したがって、効率的な推論実行が可能ないように知識ベースの形式を変換する必要があり、これを行う機能が知識ベースコンパイル機能である。したがって、D I S K E T の知識ベースには、知識ベースソースとコンパイルされた知識ベースの 2 つの形式がある。今後、前者を単に知識ベース、後者をコンパイル済知識ベースと呼ぶことにする。

1. 1. 3 ケースデータ作成・編集機能

ケースデータとは、診断対象（事故）に関連する生起事象の値の時間変化を時系列データとして図 1. 5 に示すようなリスト構造にしたものである。具体例として蒸気発生器伝熱管破損事故のケースデータを図 1. 6 に示す。推論実行時に予めケースデータがメモリー上にロードされている場合、ルールの適用等により、ある生起事象の値を参照する必要がある時、まずケースデータが参照される。もし値があれば、それが推論に使われる。ない場合は、利用者に対して質問形式で生起事象値の問い合わせが行われる。

DISKETの異常診断では、ケースデータにより事故データ（生起事象の値）の一括入力が可能であり、迅速な診断が行える。

ケースデータの作成、修正は、（IERIAS内蔵の）ケースジェネレータおよびケースエディタを用いて行うことができる。

1. 1. 4 推論機能

IERIASの推論は包括型推論機能と予測コンサルテーション型推論機能の2段構成になっている。推論の過程は、まず、警報や圧力・温度変化等の事故データを基に前向き推論を行い、確信度の高い仮説について推論結果の表示をする包括型推論を行う。次に、より正確な異常の原因の同定を行うために、包括型推論で推定された確率度の高い仮説を中心に利用者との会話型の質疑応答により予測コンサルテーション型推論を実行する。推論は、前向き推論と後ろ向き推論を併用し特定の仮説について、それを推論するに至った過程など包括型推論結果を理解する上で有用な情報の提示を行う。表 1. 1 に予測コンサルテーション型推論のメニューとその機能を示す。詳細は第3章の操作マニュアルを参照のこと。

1. 1. 5 その他の機能

(1) 計算コードの制御

計算コードを用いて診断結果から入力条件を自動的に決定し、計算実行を自動起動させる機能である。

(2) 知識ベースデバッグ機能

知識ベース作成時に有効な機能として、知識ベースデバッグ機能がある。これは知識ベースの全てのルールについて、推論に用いられたか否かを図 1. 7 に示すような形式で順次表示する。具体的には、推論を実行する前にTESTコマンドを実行し、推論終了後、予測コンサルテーションモードでTRACEコマンドを実行する。

1. 2 知識ベースの構造

D I S K E Tの知識ベースは、図1. 8に示すようにデータ定義部とルール部に大別され、前者は、原因 (HYPOTHESES) と結果 (Findings) に関する知識を含み、後者は、異常の原因と生起事象、あるいは異常の原因同志、生起事象同志の関係に関する知識をルールの形式で含んでいる。また、ルールに含まれる異常の原因、生起事象は、英数字からなる略号で表現されるため、ルールの記述が簡略化されている。更にD I S K E Tの知識ベースの特徴の一つとして、ルール部に含まれるルールを複数の知識ユニットに分割することが可能であり、不用なルールを適用することなく推論を実行でき、推論の効率化に有効である。

診断のための推論とは直接関係ないが、推論実行の前後、あるいは推論の途中で実行される種々の処理を、知識ベース内の実行処理定義部で定義することができるのも特徴の一つである。定義可能な処理は、FORTRANで書かれた外部プログラムの呼出し実行、TSSコマンドの実行、実行処理定義部で定義したLISP関数の実行である。

1. 2. 1 データ定義部

先に述べたように、データ定義部には、原因と結果に関する知識を含み、それぞれ、事故仮説 (Accident Hypothesis) と生起事象 (Finding) の二つのブロック内に、英数字からなる略号を用いて定義される。

仮説ブロックには、図1. 9に示すように、異常仮説の分類 (Taxonomy)、内部仮説 (Internal Hypothesis)、対応策仮説 (Treatment) を含んでいる。Taxonomyは、事故仮説の分類分けであり、異常仮説は、上位分類、中位分類、下位分類の階層構造に分けられる。Internal Hypotheses は、主として、推論の効率化のために導入された内部変数で、例えば推論の過程で、ある知識ユニットを適用するか否かの判断の指標として利用される。Treatment は、診断された異常仮説に対してとるべき措置を記したものであり、異常仮説や生起事象とルールで結ばれており、単なるメッセージと異なり仮説の一種 (対応仮説、Treatment Hypothesis) として取扱っている。

以上の知識は、知識ベースコンパイラによってコンパイルされ、図1. 10~1. 12に示すようなLISPのリスト構造に変換される。個々の仮説については、図1. 13に示すような属性リストが作られる。H-TYPEは、仮説の種類 (仮説分類、内部仮説あるいは対応策) を示す。LOW-LEVEL は仮説の説明文である。

生起事象ブロックには、警報 (Annunciator)、弁、ポンプの作動状況、圧力、温度等の計測値など、異常診断に利用される情報が定義される。その例を図1. 14に示す。

上記情報には、二つの種類があり、一つは、真偽 (True/False) の値を取る種類であり、MULTIPLE CHOICE, CHECKLIST および YES/NO である。これらは、警報および弁・ポンプの開閉を表わす場合に用いられる。他の一つは、量を表わす情報で、NUMERICAL である。こ

れは、圧力、温度等の計測値に用いられる。またこれらの生起事象は、一つの値だけでなく、複数の値を時系列データとして、持つことができる。これにより、推論の中で、生起事象の時間変化を考慮することが可能となっている。

以上の知識は、知識ベースコンパイラによってコンパイルされ、図 1. 15 に示すようなリスト構造に変換される。

個々の生起事象については、図 1. 16 に示すような属性リストが作られる。FIND-TYPE は生起事象の値の設定方法を示し、I, E, R (この意味については第 2 章参照) のいずれかである。

VALUE-TYPE は、生起事象の種類を示し、M (MULTIPLE CHOICE)、C (CHECKLIST)、Y (YES/NO)、および N (NUMERICAL) のいずれかである。TIME-CYCLE は、記憶される生起事象内リスト (生起事象の時間変化を表わす時系列データ) の要素数 (データ点数) であり、NAME は、生起事象の説明文、LIMIT は、" NUMERICAL " 特有の属性リストであり、生起事象の上下限値を表わす。

1. 2. 2 ルール部

ルール部に含まれる知識は、前述のデータ定義部で定義された生起事象と事故仮説との関係に関する知識であり、これをルールと呼ぶ。主要なルールには、次の 3 種類がある。

- ・ F F ルール
生起事象間の関係を表わすルール
- ・ F H ルール
生起事象と事故仮説間の関係を表わすルール
- ・ H H ルール
事故仮説間の関係を表わすルール

以上のルール以外に I F - T H E N 形式のルールがある。この形式の T H E N 部に上記 3 種類のルールの記述が可能である。但し、H H ルールは I F - T H E N 部内でのみ記述できる。I F - T H E N ルールを用いる利点は、I F 部の条件が成立した場合のみ、T H E N 部のルールが適用されるため、推論の効率化を図れる点である。各ルールの例を図 1. 20 に示す。

F F, F H, H H ルールの左辺および右辺を、それぞれ条件部、実行部と呼び、F H および H H ルールの実行部は、確信度 (C F : Certainty Factor) と呼ぶ $-1.0 \sim 1.0$ の値を持つ。正の C F は、条件部が成立した時の実行部にある仮説の確からしさの程度を、負の C F は、逆に仮説の否定の度合いを表わし、1.0 は完全肯定を、-1.0 は完全否定を意味する。

同一の仮説が複数のルールで指示される場合は、まず同符号の C F 同志を Bayse の式に基づき合成した後、合成した正負の C F を合計した値を、その仮説の C F とする。

図 1. 17 に示したルールは、コンパイル後は、図 1. 18 に示すような表現になる。

上記3ルール以外に、DISKETでは、次に示すルールが使用可能である。

- ・プログラム実行ルール

後述の実行処理定義部で定義した手続きをルールの条件が成立した時に実行する。

- ・ポーズルール

ポーズルールが適用され、条件部が満足されると、一時ルールの適用を中断し、ルールの実行部に含まれている文字（文章）が表示され、コンサルテーションが開始される。コンサルテーション終了後は、再びルールの適用が開始される。

- ・メッセージ表示ルール

このルールの条件部が満足されると実行部に含まれている文字（文章）が画面上に表示される。

上記のルールは、推論の過程では知識ユニットと呼ばれるルール集合として、一括して取り扱われる。知識ユニットには、図1. 19の例に示すように、ユニット適用の判定基準となる条件部があり、この条件が満足すれば、ユニット内の各ルールが適用される。知識ユニットの区切りは、次の知識ユニット宣言文“**KU”の直前までである。

図1. 19に示した知識ユニットは、コンパイルされて図1. 20に示すような表現になる。この知識ユニットは、“KU2”と名付けられ、ユニット名の後に、ユニットの適用条件が続く。次の行からは、図中の説明のごとくである。

1. 2. 3 実行処理定義部

実行処理定義は、推論の直前、直後あるいは、推論の途中でプログラム実行ルールが成立した場合に、外部プログラムを呼出し実行させる（アクション）手続きを定義する部分である。手続きを記述する言語はLISPあるいはFORTRANであり、LISPの外部関数呼出し機能を用いれば、TSSコマンドも処理手続きとして定義できる。図1. 21に実行処理定義の例を示す。これは“CASE-LOAD”という関数をLISPで定義しており、内容は、TSSコマンドを用いて“CASE,LOAD(TEMPNAME)”というFORTRANプログラムを実行するのに必要なデータファイルを定義し、プログラム実行後、ファイルを切離す手続き”である。

推論の実行中にルールが成立した時に、アクションを実行したい場合は、次に示すようなプログラム実行ルールを用いる。

F (ATRPRE, T) -> I (アクション名)

1. 2. 4 ルール表現における時間の取扱い

第1. 1節の冒頭で本システムの特徴の一つとして述べたように、本システムでは、生起事象発生の前後関係あるいは、時間的な履歴を考慮したルールを取扱うことが可能である。以下にそのルールの具体例を示す。

(i) 時間的前後関係を表わすルール

例えば「発電機トリップの警報 (ATRPGE) が発信された後にタービントリップの警報 (ATRPTR) が発信された場合、発電機系統に異常 (FGENES) が発生した可能性が高い (C F = 0.6)」という場合には、次のようにルール表現される。

$$F(\text{ATRPTR}, T) \text{ AFTER } F(\text{ATRPGE}, T) \rightarrow H(\text{FGENES}, 0.60)$$

上の例のように、生起事象同志をA F T E R結合して、ルールの条件部に含めることにより生起事象の発生の前後関係を推論に取込むことができる。

(ii) 時間的な履歴を考慮したルール

原子炉異常時における過渡応答では、たとえば、圧力が一時上昇したが、現在は定常圧力より低くなっているとか、ポンプが一度止ったが、現在稼働しているような場合がある。原子炉の異常診断においては、上記のような過去の状態が重要になる場合がある。D I S K E Tのルール表現では、以下の例に示すように、「かつて、給水ポンプ (PMSGFW) が停止していたらなら、タービントリップ信号が発信している」という表現が、“#”記号を用いることで可能となる。

$$F(\text{PMSGFW}, \#, F) \rightarrow F(\text{TURSTR}, T)$$

“#”記号が無い場合は、「給水ポンプが停止しているなら、タービントリップ信号が発信している」という意味になり、推論を行った時刻での生起事象の値のみ考慮される。

1. 3 推論

DISKETによる異常診断の中心は、包括型推論による異常原因の抽出である。

図1. 2. 2には、推論の流れの全体的な概略を示す。まず推論の前処理として、知識ベースのコンパイル、あるいは、コンパイル済知識ベースのローディングを行い、さらに事故データ（生起事象値）が格納されている”ケースファイル”のローディングを行う。

次に推論を行う時刻（TIME-ID）の指定を行い、生起事象値の入力方法の選択を行う。ケースファイルが予めローディングされていない時は、一問一答による生起事象値の入力を行う必要がある。ケースファイルがローディングされている場合でも、それが不完全な場合、推論実行中に値を持たない生起事象を含むルールが適用されるとその都度、値の問い合わせがある。

知識ベースの実行処理定義部において推論前アクションが定義されている場合は、それが実行された後、包括型推論が開始される。推論終了後、推論後アクションが定義されていれば、それが実行され、予測コンサルテーション型推論に移る。

1. 3. 1 包括型推論の手順

包括型推論は、知識ベース内において、”GENERAL KB”として定義されるルール群を用いて行う。ルールの適用の順序は、原則的には、知識ベース内に記述されている順であるが、DISKETシステムでは、1. 2節でも述べたように、知識をユニット化することにより、不用なルールの適用を避け、推論の効率化を図っている。図1. 2. 3に知識ユニットを用いた場合の包括型推論の流れを示す。

知識ユニットは、知識ベースコンパイル時に知識ベース内に記述されている順に番号が付けられ、1番目の知識ユニットから順に適用が試みられる。既に適用された知識ユニットはパスされ、未適用の場合は、そのユニットの適用条件がチェックされ、それが成立しなければ、次の知識ユニットに移り、成立していればその知識ユニット内のルールを適用する。ルールはFFルール、FHルール、HHルールの順に適用される。そのなかに成立するルールがあれば、次の知識ユニットに移らず、第1番目の知識ユニットに戻り、ルール成立の再チェックを行う。これは、あるルールの成立により、以前成立していなかったルールが成立する場合があるからである。最後の知識ユニットをチェックして、これが適用済か、または条件部が成立していなければ、他に適用すべき知識ユニットがないので、包括型推論を終了する。

次に各知識ユニット内でのルールの適用の手順について説明する。

ルールは1. 2. 2節でも説明したように、条件部と実行部から成り、まず条件部が評価され、条件部が成立すれば実行部が実行され、不成立であれば何も行わない。条件部を評価するに際して、そこに含まれる生起事象の値、あるいは仮説の確信度が未決定の場合、次の手順に従い、生起事象値あるいは確信度を求める。

a) 仮説の確信度

その仮説を実行部に含むルールの成立の有無を調べる。

b) Rタイプの生起事象値

”R”タイプの生起事象は、ルールにより値が与えられるので、その生起事象を実行部に含むルールの成立の有無を調べる。

c) Eタイプの生起事象値

”E”タイプの生起事象は、推論実行中にユーザーにより値が与えられる。そこで、その生起事象値をユーザーに対して問い合わせのメッセージが画面上に出力される。ユーザが問い合わせに対して値を入力すると、それが生起事象の値として使用される。

d) Iタイプの生起事象値

”I”タイプの生起事象は、通常、包括型推論実行前にケースファイルから一括入力により値を与えられる。従って、ルール適用時に値が無ければ、ルールは単に不成立とする。RタイプやEタイプの生起事象の場合のようにその値を得るために他のルールの適用、あるいはユーザーへの質問は行わない。

a) あるいは b) によるルール適用の条件部評価中に値を持たない生起事象あるいは確信度を持たない仮説が出現した時は、更に a)～b) が行われる。またこれらのルール適用は同一知識ユニット内のルールについて行われる。

また値不明生起事象および仮説は、同一知識ユニット内のそれぞれFFルール、FHルール、またはHHルール内でそれを条件部に持つルールを探す。

知識ユニット内の個々のルールの適用順序は基本的には知識ベースに記述されている順に適用される。ただし a) あるいは b) のように、あるルールの条件部評価中に他のルールが適用されることがあり、一度適用されたルールは再び適用されることはない。

1. 3. 2 確信度

本システムでは推論における曖昧さを表す尺度として、確信度 (Certainty Factor ; CF) を用いており、その計算方法は基本的にMYCINにおける計算方法に準じている。

本システムで用いているCFの特徴は、

- i) その値の範囲が $-1.0 \sim 0.0$ あるいは $0.0 \sim 1.0$ である。
- ii) 正のCFは、その仮説の確からしさの度合いを表わし、 $CF=1.0$ は、完全肯定を表わす。逆に負のCFは、その仮説の否定の度合いを表わし、 $CF=-1.0$ は、完全否定である。
- iii) 一つの仮説について、複数のルールによってCFを与えることができ、それらは、Bayesの定理に基づき合成され、その値はルールの成立順序に依存しない。
- iv) iii)の特徴により、情報の欠落による診断不能ということがない。

などである。

以下に確信度の合成計算について説明する。

MB (H, F) を生起事象 F に基づく仮説 H に関する確信度の増分とし、MD (H, F) を不信度の増分とする。MB, MD は条件付および先験確率の形式では、

$$MB(H, F) = \begin{cases} 1 & : P(H)=1 \\ \frac{\text{MAX} [P(H|F), P(H)] - P(H)}{1 - P(H)} & : \text{その他の場合} \end{cases}$$

$$MD(H, F) = \begin{cases} 1 & : P(H)=0 \\ \frac{\text{MIN} [P(H|F), P(H)] - P(H)}{P(H)} & : \text{その他の場合} \end{cases}$$

(P(H) は、仮説に対する先験確率を示す。)

と表現される量である。個々のルールで与えられる確信度は、ルールの条件部が含む生起事象の集合を F と置換すれば、正負に従ってそれぞれ MB, MD に対応する。

さて、複数のルール R₁, R₂, …, R_n に対して成立する仮説 H に対して、以下の方式で段階的に MB, MD を計算する。

$$MB(H, R_i \& R_j) = \begin{cases} \text{if } MD(H, R_i \& R_j) = 1 : 0 \\ \text{otherwise} \\ MB(H, R_i) + MB(H, R_j) [1 + MB(H, R_i)] \end{cases}$$

$$MD(H, R_i \& R_j) = \begin{cases} \text{if } MB(H, R_i \& R_j) : 0 \\ \text{otherwise} \\ MD(H, R_i) + MD(H, R_j) [1 - MD(H, R_i)] \end{cases}$$

また、あるルール R_i が真であると確かに知られていないが、それまでの経験から確信度が確立しているものとする。このとき、MB'(H, R_i), MD'(H, R_i) がそれぞれ R_i が真であると仮定した場合の確信及び不信の程度を表わすとすれば、実際の確信度、不信度の増分は

$$MB(H, R_i) = MB'(H, R_i) \cdot \text{Max} [0, CF(R_i, E)]$$

$$MD(H, R_i) = MD'(H, R_i) \cdot \text{Max} [0, CF(R_i, E)]$$

(ここでEは、それぞれの経験全体の集合を示す。)

で与えられる。

事象が発生した場合、該当するすべてのルールについて上記方式で結合計算した結果の確信度の増分、不信度の増分をそれぞれ MB^* 、 MD^* とすると、ある事故仮説 H_i に対する確信度(確信測度、又は同定測度ともよばれる) CF は

$$CF(H_i) = MB^*(H_i) - MD^*(H_i)$$

で計算を実行される。

表 1. 1 予測コンサルテーションメニュー

メニュー	機能
BACK	ルールのバックトレース
CR	排反ルールの出力
CRF	排反生起事象の出力
DETAIL	予測コンサルテーション型推論
DR	包括型推論結果の再出力
FIND	生起事象の値の出力
FIX	生起事象の値を与える
HELP	HELPメッセージ出力
HYPO	推論過程の表示
NR	否定ルールの出力
PC	予測コンサルテーション出力コントロール用 パラメータの変更
PRED	シミュレーションプログラムの起動
PDX	最新の推論結果の出力
QUIT	予測コンサルテーションの終了
RECONF	生起事象の再確認
RED	ルールの部分修正
RFIND	指定した生起事象を含むルールの出力
RFINDC	指定した生起事象を条件部に持つルールの出力
RULE	ルールの内容説明
STATUS	DISKETのシステムステータスの表示
SUM	全生起事象の出力
TRACE	ルールのトレース
UF	不明生起事象の出力

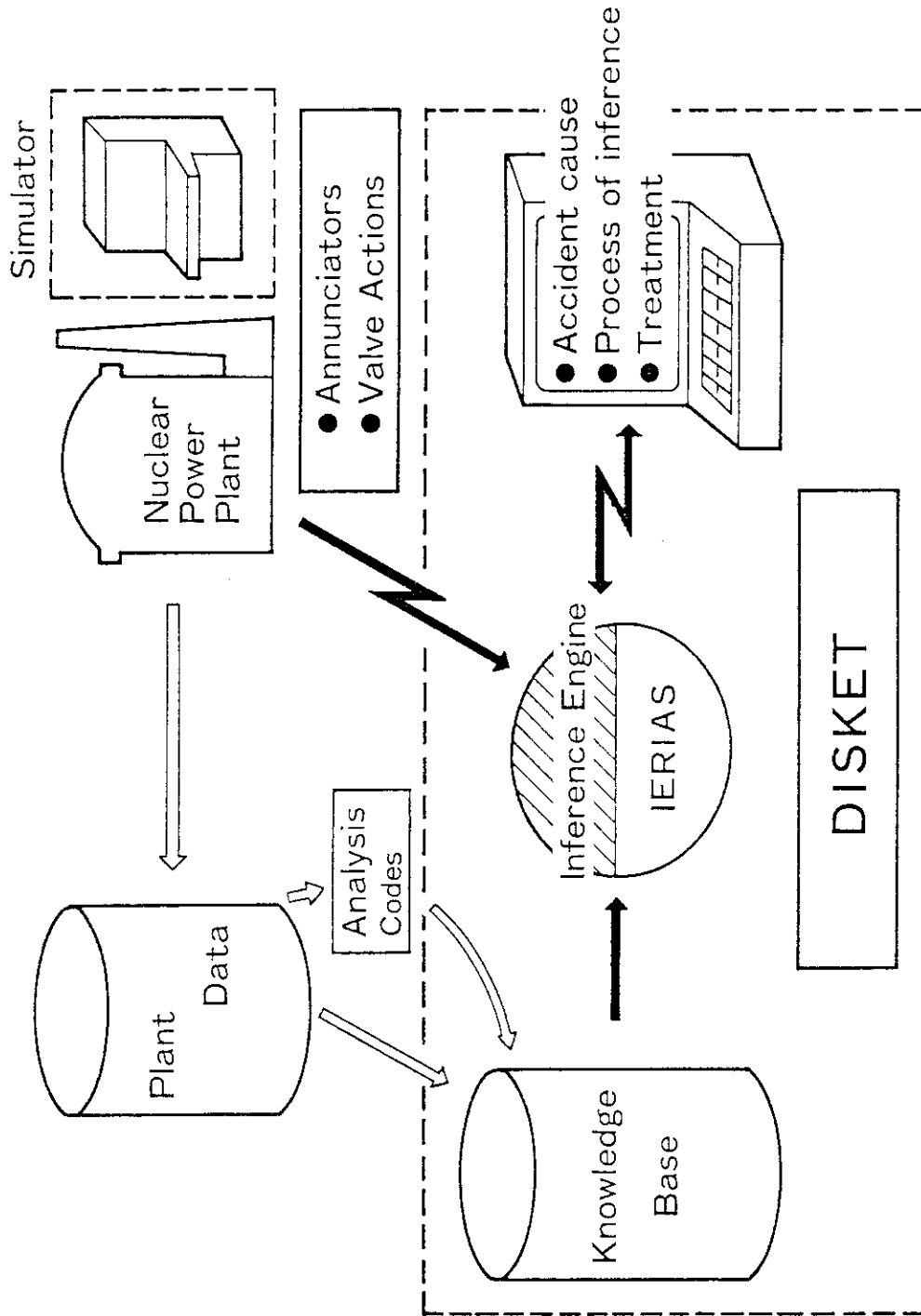


図1.1 DISKETシステムの概念図

```

FPCSPM(CF=NIL)
| ---- <1>
| --- HS2-1-1(0.60)
|   --- COND
|     --- FPCS(0.10 1.0)
|       --- <1>
|         --- FH1-2(0.50)
|           | --- COUNT 1
|             | --- SRCSL1(TRUE)
|             | --- SRCSL2(TRUE)
|         --- FH1-4(0.50)
|           | --- COUNT 1
|             | --- FLRCS1((-50 50))
|             | --- FLRCS2((-50 50))
|             | --- FLRCS3((-50 50))
|         --- FH1-5(0.45)
|           | --- APRECON(TRUE)
|         --- FH1-7(0.45)
|           | --- ARADMAN(TRUE)
|         --- FH1-8(0.40)
|           | --- ESSACT(TRUE)
|         --- FH1-10(0.40)
|           | --- COUNT 1

```

図 1. 2 決定木の出力例

知識ベース内でのルール表現

FH-9 F(AVEVPRL,T)&F(APREPRL,T) → H(FPCS,0.30)

HS2-1-1 (1:F(FLRCST,0:90).F(FLRCS2,0:90).F(FERS3,0:90).F(AFLRCSL,1)) →
H(FPCSPM,0.60)

ルール説明機能を用いた時のルール表現

FH1-9 F(PRESSURIZER LEVEL LOW IS TRUE) AND F(PRESSURIZER PRESSURE LOW
IS TRUE → H(FAILURE OF PRIMARY COOLANT SYSTEM IS WITH CF=0.30)

HS2-1-1 (1 OUT OF F(REACTOR COORANT FLOW NO.1 LOOP (%) IS ABOVE 0 AND
BELOW 90).F(REACTOR COOLANT FLOW NO.2 LOOP (%) IS ABOVE 0 AND
BELOW 90).F(REACTOR COOLANT FLOW NO.3 LOOP (%) IS ABOVE 0 AND
BELOW 90).F(LOW REACTOR COOLANT FLOW IS TRUE)) → H(FAILURE OF
PRIMARY COOLANT PUMP IS WITH CF=0.60)

図 1. 3 ルール説明機能の出力例

```

* ENVIRONMENT* = FPCS      ==>
FPCS(CF=NIL) -- <1> -- FH1-2(0.50) --- COUNT 1 ----- SRCSL1(TRUE)
|
| - SRCSL2(TRUE)
| - FH1-4(0.50) --- COUNT 1 ----- FLRCS1(-50 50)
|
| - FLRCS2((-50 50))
| - FLRCS3((-50 50))
| - FH1-5(0.45) --- APRECON(TRUE)
| - FH1-7(0.45) --- ARADMAL(TRUE)
| - FH1-8(0.40) --- ESSACT(TRUE)
| - FH1-10(0.40) -- COUNT 1 ----- PRCONT((0.65 *))
|
| - TECONT((50 *))
| - FH1-6(0.35) --- AFLCHAH(TRUE)

*****

SPRZPL (PRESSURIZER LOW PRESSURE) :
SRCSL1 (RCS LOW FLOW 13) :
SRCSL2 (RCS LOW FLOW 23) :
AFLRCOL (LOW REACTOR COOLANT FLOW) :

```

図 1. 4 画面二分割

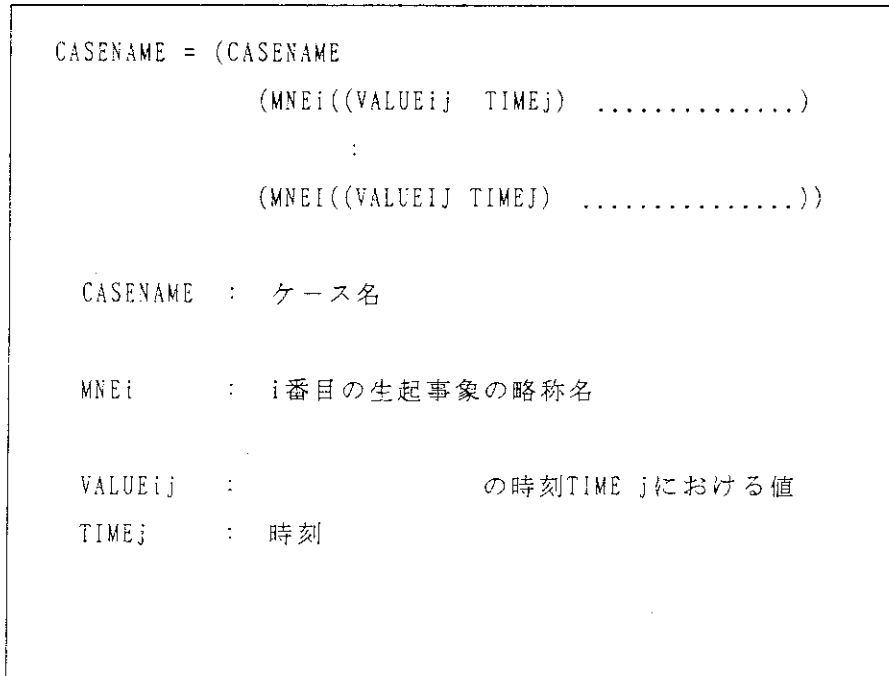


図 1. 5 ケースデータの構造

```

(SETQ ##CASE 'CASE15)
(SETQ CASE15 '(CASE15 (NPPA ((T 800)(F 0)))
(OHA ((F 000)))
(ATRPRE ((T 831.36)(F 0)))
(ATRPTR ((T 831.36)(F 0)))
(ATRPGE ((T 832.16)(F 0)))
(ACRDSTP ((F 000)))
(ACRDBTM ((T 831.36)(F 0)))
(ACRDWIL ((T 819.20)(F 0)))
(ALEVPRH ((F 000)))
(ALEVPRL ((T 801.00)(F 0)))
(APREPRH ((F 0)))
(APREPRL ((T 802.20)(F 0)))
(ATEPRLH ((F 000)))
(ATEPRTH ((F 000)))
:
(FLRCS1 ((100 000)))
(FLRCS2 ((100 000)))
(FLRCS3 ((100 000)))
(PREPRZ ((160 850)(153 830)(162 810)(160 0)))
(LEVPRZ ((0 840)(0 820)(45 0)))
(LEVPRT ((0 000)))
(TEMPRL ((50 000)))
(TEMPRT ((25 000)))
(PREPRT ((+0.1000000 +00 000)))
(TEMHL1 ((286 840)(322 0)))
(TEMHL2 ((286 840)(322 0)))
(TEMHL3 ((286 840)(322 0)))
(TEMCL1 ((285 840)(287 0)))
(TEMCL2 ((285 840)(287 0)))
(TEMCL3 ((285 840)(287 0)))
(TEMAVE ((285 840)(304 0)))
(TEMDEV ((0.7 840)(0 0)))
(QFLUX ((0 834)(84 820)(100 0)))
(LCRODA ((0 831)(220 0)))
(LCRODB ((0 831)(220 0)))
:
(VF498 ((F 833.04)(T 0)))
(VM151A ((F 000)))
(VM151B ((F 000)))
(VM151C ((F 000)))
(VL102A ((F 000)))
(VL123A ((F 000)))
(YREVFL1 ((N 000)))
(YREVFL2 ((N 000)))
(YREVFL3 ((N 000)))
(YTEMCH1 ((N 000)))
(YTEMCH2 ((N 000)))
(YTEMCH3 ((N 000))))

```

図 1. 6 ケースデータの例


```

** IERIAS RULE APPLIED HISTORY **
FF1-1  -1  (NONAPPLIED)
          FF1-1      :F(PRCONT,0.62:*) -> F(AINISG,T)
FF1-2   1  (APPLIED)
          FF1-2      :F(LCRODD,0:10) -> F(ACRDBTM,T)
FF1-3  -1  (NONAPPLIED)
          FF1-3      :F(LORODD,220:*) -> F(ACRDWIL,T)
FF1-4   1  (APPLIED)
          FF1-4      :F(LEVPRT,0.05:*) -> F(ALEVPRH,T)
FF1-5  -1  (NONAPPLIED)
          FF1-5      :F(LEVPRT,-1:-0.06) -> F(ALEVPRL,T)
FF1-6  -1  (NONAPPLIED)
          FF1-6      :F(PREPRZ,163:*) -> F(APREPRH,T)

```

図 1. 7 トレース実行例

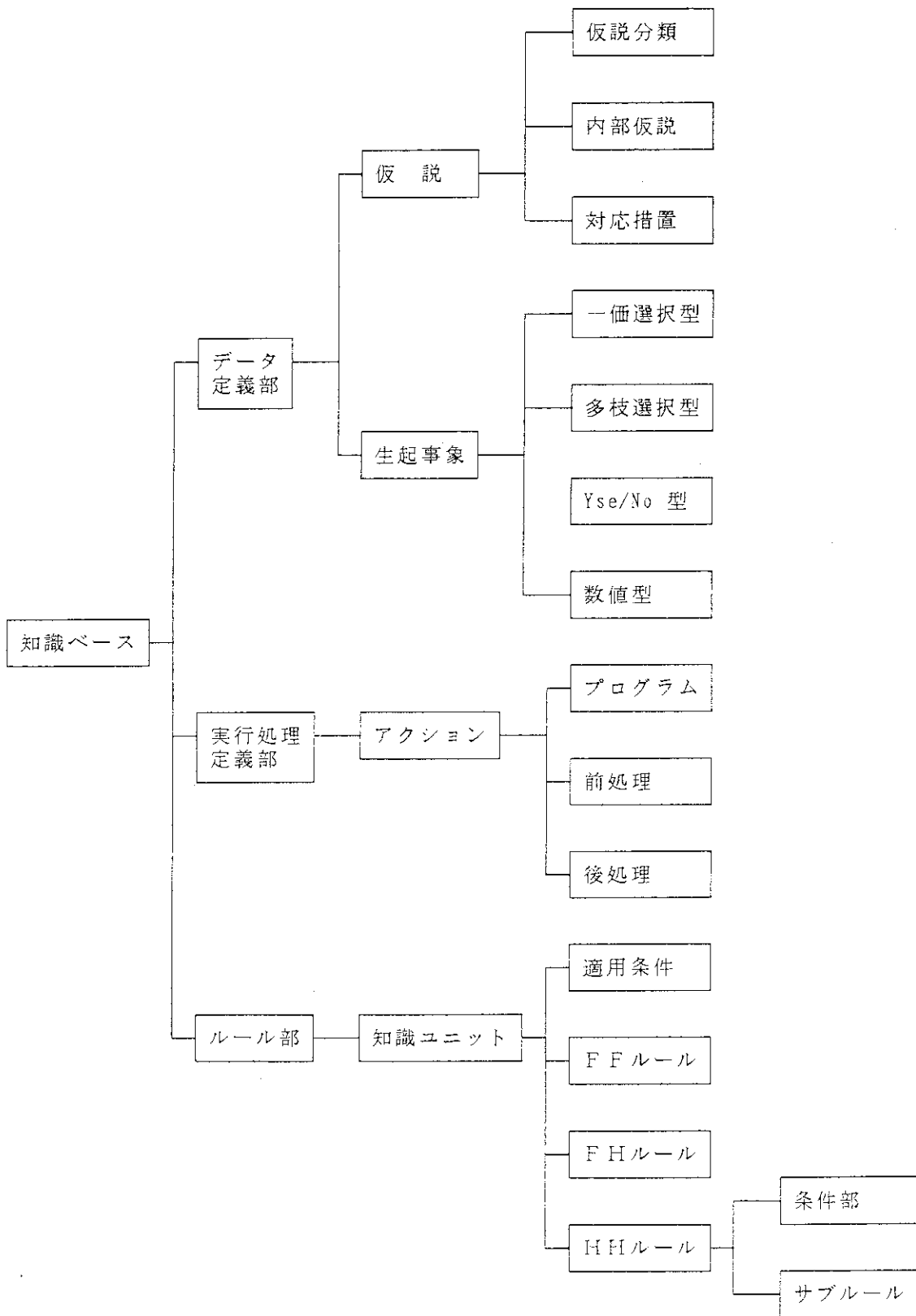


図 1. 8 知識ベースの構造

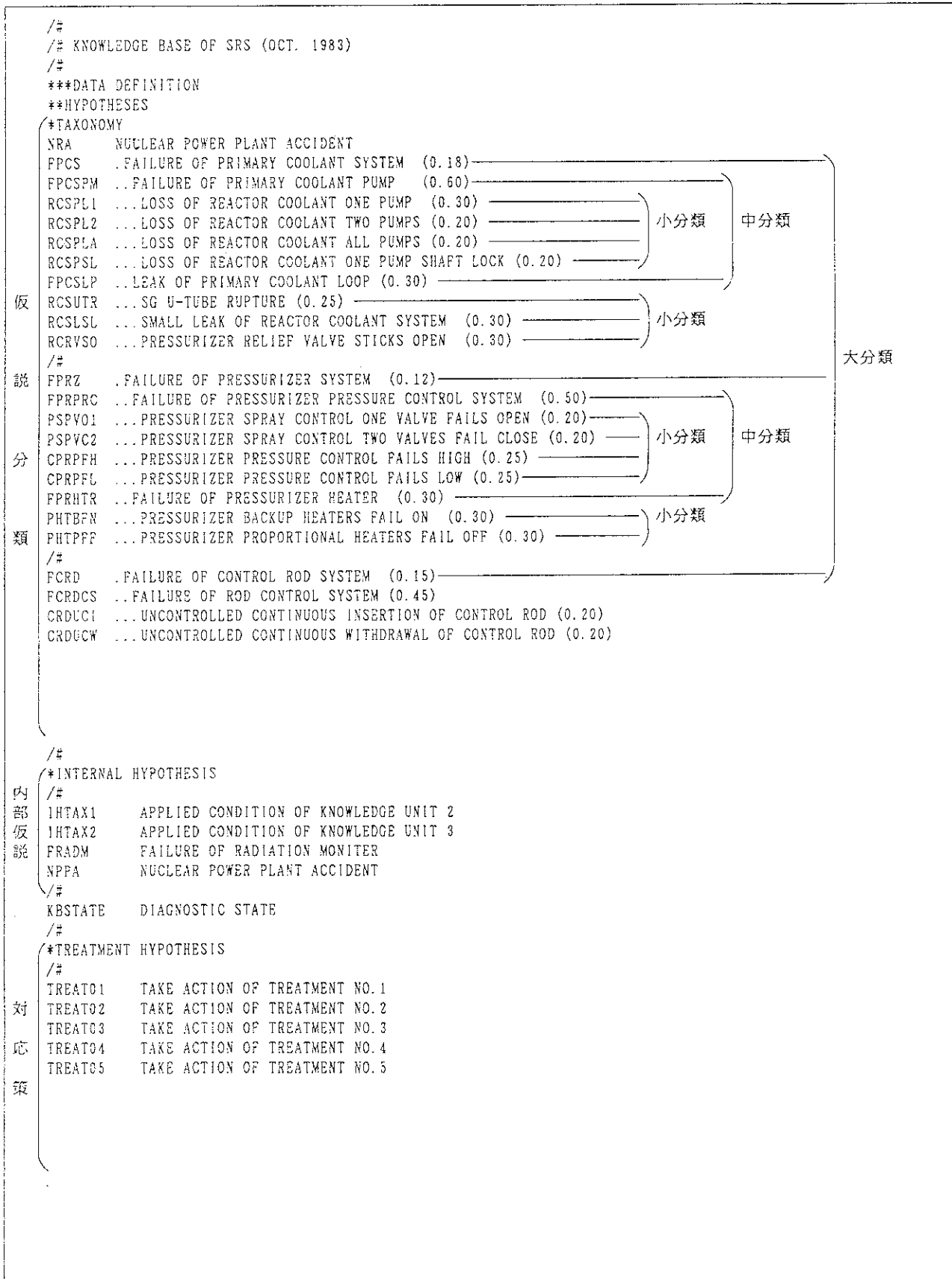


図 1. 9 知識ベース内の仮説ブロック

```
(TAXONOMY
(NRA (FPCS (FPCSPM (RCSPL1) (RCSPL2) (RCSPLA) (RCSPSL))
      (FPCSLP (RCSUTR) (RCSLSL) (RCRVSO)))
(FPRZ (FPRPRC (PSPV01) (PSPV02) (CPRPFH) (CPRPFL))
      (FPRHTR (PHTBFN) (PHTPFF)))
(FCRD (FCRDCS (CRDUCI) (CRDUCW) (CRDMDF) (CRDATF) (CLATEH))
      (FCRDOM (CRDDR1) (CRDEJ1) (CRDWDF) (CRDMAS) (CRDMIS)))
(FCVCS (FCVCLC (CVTLFH) (CVTLFL)) (FCVCCG (CHAGFL)))
(FSTM (FSTMPC (CSHPFH) (CSHPFL) (CEHCAL)) (FSTMLL (SPVLA) (SGSLL)))
(FTRG (FTURCS (TGVF01) (TGVFC1) (TURRBF) (TURTRP)) (FGENES (GENTRP)))
(FCOND (FCONCW (CCWPL) (CCWTL)))
(FSGFW (FSGLCN (CSCLFH) (CSGLFL)) (FSGFWL (FWFPL) (FWFTL) (HPHDPL)))
```

図 1. 10 コンパイル後の仮説分類の構造

```
(TREATMENT
(TREAT01)
(TREAT02)
(TREAT03)
(TREAT04)
(TREAT05)
(TREAT06)
(TREAT07)
(TREAT08)
(TREAT09))
```

図 1. 11 コンパイル後の対応措置の構造

```
(INTER-HYPO (IHTAX1) (IHTAX2))
```

図 1. 12 コンパイル後の内部仮説の構造

```
(H-TYPE TAXONOMY
LOW-LEVEL
((FPCSLP +0.3000000--+00) (FPCSPM +0.6000000--+00))
NAME
"FAILURE OF PRIMARY COOLANT SYSTEM")
```

図 1. 13 コンパイル後の仮説の構造

```

/#
**FINDINGS
*BEGIN QUESTIONNAIRE
/#
/# ANNUNCIATORS
/#
*CHECKLIST
ANNUNCIATORS OF TRIP:
ATRPRE REACTOR TRIP
ATRPTR TURBINE TRIP
ATRPGE GENERATOR TRIP
*CHECKLIST (2)
ANNUNCIATORS OF CONTROL ROD:
ACRDSTP ROD STOP
ACRDBTM ROD BOTTOM
ACRDWIL ROD WITHDRAW LIMIT
/#
*CHECKLIST (5)
ANNUNCIATORS OF PRESSURIZER:
ALEVPRH PRESSURIZER LEVEL HIGH
ALEVPRL PRESSURIZER LEVEL LOW

/#
/# PUMP
*CHECKLIST (3)
PUMPS:
PMRCP1 REACTOR COOLANT PUMP 1A RCP-1A
PMRCP2 REACTOR COOLANT PUMP 2A RCP-2A
PMRCP3 REACTOR COOLANT PUMP 3A RCP-3A
PMCHAG CHARGING PUMP 1-CH-P1A

/#
/# VALVE
/#
/# OPEN... TRUE CLOSE... FALSE
/#
*CHECKLIST (5)
VALVE POSITION:
VP455A PRESSURIZER SPRAY VALVE A PCV 455A
VP455B PRESSURIZER SPRAY VALVE C PCV 455B

/#
/# YES//NO
*YESNO (5)
YREVFL1 REVERSE FLOW LOOP 1
*YESNO (5)
YREVFL2 REVERSE FLOW LOOP 2
*YESNO (5)
YREVFL3 REVERSE FLOW LOOP 3

/#
/# NUMERICAL
/#
*NUMERICAL (0 150) (5 1)
FLRCS1 REACTOR COOLANT FLOW NO. 1 LOOP (%)
*NUMERICAL (0 150) (5 1)
FLRCS2 REACTOR COOLANT FLOW NO. 2 LOOP (%)
*NUMERICAL (0 150) (5 1)
FLRCS3 REACTOR COOLANT FLOW NO. 3 LOOP (%)
*NUMERICAL (7 1)
GFRCS1 GRADIENT OF COOLANT FLOW NO. 1 LOOP (%)

```

図1. 14 知識ベース内の生起事象ブロック

```

(FINDINGS
  (TYPE-OF-PROBLEM (NPPA) (OTHA))
  (ANNUNCIATORS-OF-TRIP (ATRPRE) (ATRPTR) (ATRPGE))
  (ANNUNCIATORS-OF-CONTROL-RCD (ACRDSTP) (ACRDDTM) (ACRDWIL))
  (ANNUNCIATORS-OF-PRESSURIZER
    (ALEVPRH)
    (ALEVPRL)
    (APREPRH)
    (APREPRL)
    (ATEPRLH)
    (ATEPRTH)
  (ANNUNCIATORS-OF-PRIMARY-LOOP (AFLRCSL) (ATEAVRF) (ATEAVGH) (ATEAVGL))
  (ANNUNCIATORS-OF-CVCS
    (APREVTH)
    (APREVTL)
    (ATEMVTH)
    (ATEMLDH)
    (AFLCHAH)
    (AFLCHAL)
    (ALEVRWL)))

```

図 1. 15 生起事象の構造

```

(D*SETPLIST
  'FLRCSI
  '(FIND-TYPE
    I
    VALUE-TYPE
    N
    TIME-CYCLE
    5
    NAME
    "REACTOR COOLANT FLOW NO.1 LOOP (%)"
    LIMIT
    (0 150)))

```

図 1. 16 コンパイルされた生起事象

F F ルール

```

F(PRCNT , #, 0.62 : * ) -> F(AINISG , T )
F(LCRODD , #, 0 : 10 ) -> F(ACRDBTM , T )
F(LCRODD , #, 220 : * ) -> F(ACRDWIL , T )

```

F H ルール

```

(1 : F(APREPRH, T),F(APREPRL, T)) -> H(FPRZ , 0.15)
(1 : F(ALEVPRH, T),F(ALEVPRL, T)) -> H(FPRZ , 0.20)
(1 : F(ATEPRLH, T),F(ATEPRTH, T),F(APRERTH, T)) -> H(FPRZ , 0.15)
F(SPRFXH , T ) -> H(FCRD , 0.30)
F(ACRDWIL, #, T ) -> H(FCRD , 0.30)
F(ATEAVRF, #, T ) -> H(FCRD , 0.05)

```

I F - T H E N ルール (H H ルール)

```

*IF
H(FPCS , 0.1:1.0)
*THEN
(1 : F(FLRCS1,0:90),F(FLRCS2,0:90),F(FLRCS3,0:90),F(AFLRCSL,T))
-> H(FPCSPM,0.6)
(1 : F(AFLCHAH,T),F(FLCHAG,20:*),F(APREVTL,T),F(VL1460A,F))
-> H(FPCSLP,0.5)
F(AFLRCSL , F ) -> H(FPCSPM , -0.50)
F(ACRDWIL , T ) -> H(FPCSLP , 0.20)
F(ALEVPRL , T ) & F(APREPRL,T) -> H(FPCSLP , 0.30)
F(PMCNSP , T ) -> H(FPCSLP , 0.50)
F(ESSACT , T ) -> H(FPCSLP , 0.30)
F(ARADMAL , T ) & H(FRADM,*:0.0) -> H(FPCSLP , 0.40)

```

図 1. 17 D I S K E T システムのルール例

コンパイル後の F F ルール

```
(SETQ FF1-1 '(((F PRCONT # +0.6200000 +00 *)) (F AINISG T)))
(SETQ FF1-2 '(((F LCRODD # 0 10)) (F ACRDBTM T)))
(SETQ FF1-3 '(((F LCRODD # 220 *)) (F ACRDWIL T)))
```

コンパイル後の F H ルール

```
(SETQ FH1-16 '(((1 (F APREPRH T) (F APREPRL T))) (H FPRZ +0.1500000 +00)))
(SETQ FH1-17 '(((1 (F ALEVPRL T) (F ALEVPRH T))) (H FPRZ +0.2000000 +00)))
(SETQ FH1-18 '(((1 (F ATEPRHL T) (F ATEPRTH T) (F APRERTH T)))
(H FPRZ +0.1500000 +00)))
(SETQ FH1-19 '(((F SPRFXH T)) (H FCRD +0.3000000 +00)))
(SETQ FH1-20 '(((F ACRDWIL # T)) (H FCRD +0.3000000 +00)))
(SETQ FH1-21 '(((F ATEAVRF # T)) (H FCRD +0.5000000 -01)))
```

コンパイル後の I F - T H E N ルール

(HH-RULES)

```
(HH2-1 (CONDITION (H FPCS +0.1000000 +00 +0.1000000 +01))
(RULE-SEQ 8)
(SUB-RULES HS2-1-1 HS2-1-2 HS2-1-3 HS2-1-4 HS2-1-5
HS2-1-6 HS2-1-7 HS2-1-8))
```

```
(SETQ HS2-1-1 '(((1 (F FLRCSI 0 90) (F FLRCS2 0 90) (F FLRCS3 0 90)
(F AFLRCSL T)))
(H FPCSPM +0.6000000 +00)))
(SETQ HS2-1-2 '(((1 (F AFLCHAH T) (F FLCHAG 20 *) (F APREVTL T) (F VL1460A
(H FPCSLP +0.5000000 +00)))
(SETQ HS2-1-3 '(((F AFLRCSL F)) (H FPCSPM -0.5000000 +00)))
(SETQ HS2-1-4 '(((F ACRDWIL T)) (H FPCSLP +0.2000000 +00)))
(SETQ HS2-1-5 '(((F ALEVPRL T) (F APREPRL T)) (H FPCSLP +0.3000000 +00)))
(SETQ HS2-1-6 '(((F PMCNSP T)) (H FPCSLP +0.5000000 +00)))
(SETQ HS2-1-7 '(((F ESSACT T)) (H FPCSLP +0.3000000 +00)))
(SETQ HS2-1-8 '(((F ARADMAL T) (H FRADM * +0.0000000 +00)))
```

*の部分は、コンパイル後の知識ユニットの一部に含まれる。

図 1. 18 コンパイル後の D I S K E T システムのルール表現

```

#
**KU
*COND H(IHTAX1, 0.4: 1.0)
#
*HH RULES
#
# NO.1    FPCSPM,FPCSLP
#
*IF
H(FPCS , 0.1:1.0)
*THEN
(1 : F(FLRCS1,0:90),F(FLRCS2,0:90),F(FLRCS3,0:90),F(AFLRCSL,T))
-> H(FPCSPM,0.6)
(1 : F(AFLCHAH,T),F(FLCHAG,20:*),F(APREVTL,T),F(VL1460A,F))
-> H(FPCSLP,0.5)
F(AFLRCSL , F ) -> H(FPCSPM , -0.50)
F(ACRDWIL , T ) -> H(FPCSLP , 0.20)
F(ALEVPRL , T ) & F(APREPRL,T) -> H(FPCSLP , 0.30)
F(PMCNSP , T ) -> H(FPCSLP , 0.50)
F(ESSACT , T ) -> H(FPCSLP , 0.30)
F(ARADMAL , T ) & H(FRADM,*:0.0) -> H(FPCSLP , 0.40)
*END
#

```

図 1. 19 知識ユニットの例

```

(SETQ KU2 '(KU2 (UNIT-CONDITION ((H IHTAX1 +0.4000000 +00 +0.1000000 +01)))
  (MESSAGE-OPTION) ←----- メッセージルールの有無
  (RULE-SEQ 0 0 9) ←----- FF, FH, HHルールのそれぞれの数
  (FF-RULES) ←----- FFルールがあればこのあとにFFルール名が続く
  (FH-RULES) ←----- FHルールがあればこのあとにFHルール名が続く
  (ACTION-INDEX) ←----- FF, FHルールの実行部に含まれるHまたはF名
  (IF-INDEX) ←----- FF, FHルールの条件部に含まれるHまたはF名
  (IN-EXCLUDE)
  (HH-RULES ←----- HHルール名      適用条件
    (HH2-1 (CONDITION (H FPCS +0.1000000 +00 +0.1000000 +01))
      (RULE-SEQ 8) ←----- HHルールに含まれるサブルールの数
      (SUB-RULES HS2-1-1 HS2-1-2 HS2-1-3 HS2-1-4 HS2-1-5
        HS2-1-6 HS2-1-7 HS2-1-8) ←----- サブルール名
      (ACTION-INDEX ←----- サブルールの実行部に含まれるH
        (FPCSPM HS2-1-1 HS2-1-3)                またはF名
        (FPCSLP HS2-1-2 HS2-1-4 HS2-1-5 HS2-1-6 HS2-1-7
          HS2-1-8))
      (IF-INDEX ←----- サブルールの条件部に含まれるHまたはF名
        (FLRCS1 HS2-1-1)
        (FLRCS2 HS2-1-1)
        (FLRCS3 HS2-1-1)
        (AFLRCSL HS2-1-1 HS2-1-3)
        (AFLCHAH HS2-1-2)
        (FLCHAG HS2-1-2)
        (APREVTL HS2-1-2)
        (VL1460A HS2-1-2)
        (ACRDWIL HS2-1-4)
        (ALEVPRL HS2-1-5)                H : 仮説, F : 生起事象
        (APREPRL HS2-1-5)
        (PMCNSP HS2-1-6)
        (ESSACT HS2-1-7)
        (ARADMAL HS2-1-8)
        (FRADM HS2-1-8))
      (IN-EXCLUDE))

```

図 1. 20 コンパイル後の知識ユニット

```
#
***ACTION DEFINITION
**ACTION CASE-LOAD
*LANGUAGE L I
(DEFUN CASE-LOAD NIL
  (CALL 'ALLOC "DA(HF.DATA) F(FT02F001)")
  (CALL 'ALLOC "DA(IERIAS.DATA) F(FT01F001)")
  (CALL 'ALLOC "DA(IECASEF.DATA) F(FT03F001)")
  (CALL 'CALL "CASEMAKE.LOAD(TEMPNAME)")
  (CALL 'FREE "F(FT01F001)")
  (CALL 'FREE "F(FT02F001)")
  (CALL 'FREE "F(FT03F001)")
  (EXFILE "IECASEF.DATA")
  (D*IE-TABLE NIL))
(CASE-LOAD
```

図 1 . 2 1 実行処理定義の例

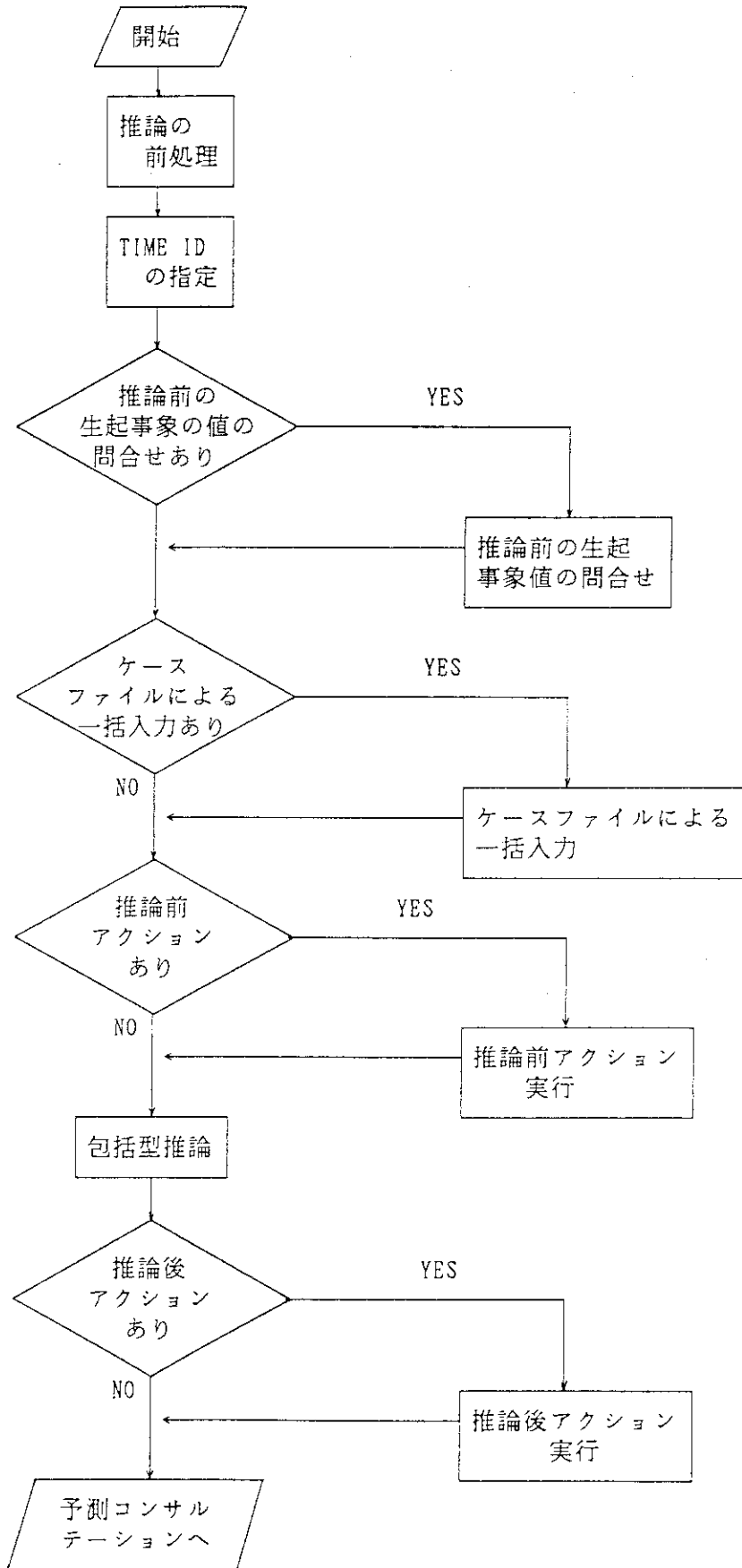


図 1. 2 2 推論の流れ

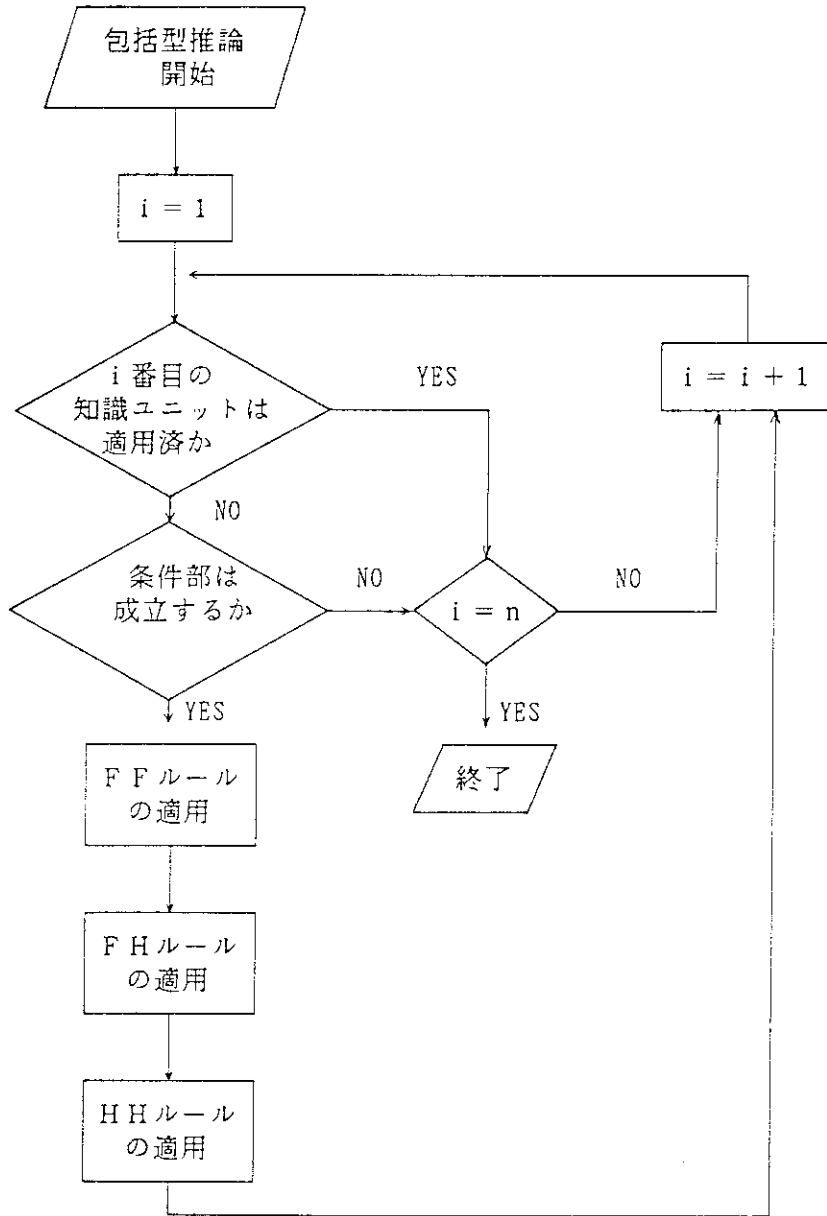


図1. 23 包括型推論におけるルールの適用手順

2. 知識ベース作成手引き

本章では知識ベースの作成について解説する。説明には、できるだけ図を用いているが、付録Cに載せた知識ベース例も参照されたい。

2. 1 知識表現

2. 1. 1 知識ベースの構造

知識ベースはデータ定義部、実行処理定義部、ルール部から構成されている。

i) データ定義部

仮説、内部仮説、対応策および生起事象を定義する。

ii) 実行処理定義部

推論の実行前後あるいは実行中に起動させるアクション（LISPプログラム、FORTRANプログラム等）を定義する。

iii) ルール部

大きく、包括推論型知識ベースと予測コンサルテーション型知識ベースに分かれ、それぞれ複数の知識ユニットを定義できる。各知識ユニットごとに、FDルール、FHルール、HHルールに分けてルールを定義する。

2. 1. 2 知識ベースの記述形式

知識ベースの各部（データ定義部、実行処理定義部、ルール部、およびそれらのより細かい分類）を区別するために、知識ベースを記述するときは、各々の定義の先頭に宣言文（ヘッダー）をかく。

宣言文の先頭には、***、**、*のいずれかが書かれる。*の数が多いほど上位の区分である。

図2. 1に知識ベースの記述形式を示す。宣言文のフォーマットのみ示したが、実際の知識ベースの内容は、点線部に記述される。

2. 1. 3 使用文字の制約

知識ベースを記述するとき、表2. 1に示す文字はその使用に制約を受ける。

表 2. 1 文字の使用制限

文字	呼 称	処 置
:	コロン	・空白におきかえられる。 ・仮説、事象等の略称に記述してはならない。
,	カンマ	・空白におきかえられる。 ・仮説、事象等の略称に記述してはならない。
&	アンパサンド	・” & ”におきかえられる。 ・仮説、事象等の略称に記述してはならない。
;	セミコロン	・以後を注釈行として扱う。
(左カッコ	・対応する右カッコがあるまで読みつづける。 ・仮説、事象等の略称に記述してはならない。
)	右カッコ	・右カッコのみ出現する時は無視される。
#	パーセント	・1カラム目にある場合は注釈行と見なし、略称に含む#のまへの文字列は知識ベース略称と見なされる。 ・1カラム目以外で使用した場合は、通常の文字セットと見なされる。
*	アスタリスク	・知識ベースの第一カラムにある時、その行は定義宣言文と見なす。
'	クォート	・(QUOTE ~)に変換する。 ・仮説、事象等に記述してはならない。
”	ダブルクォート	・対応するダブルクォートまでを一つの文字列と見なす。 ・仮説、事象等の略称に記述してはならない。
/	スラッシュ	・特殊な入力に用いる。 ・仮説、事象等の略称に記述してはならない。

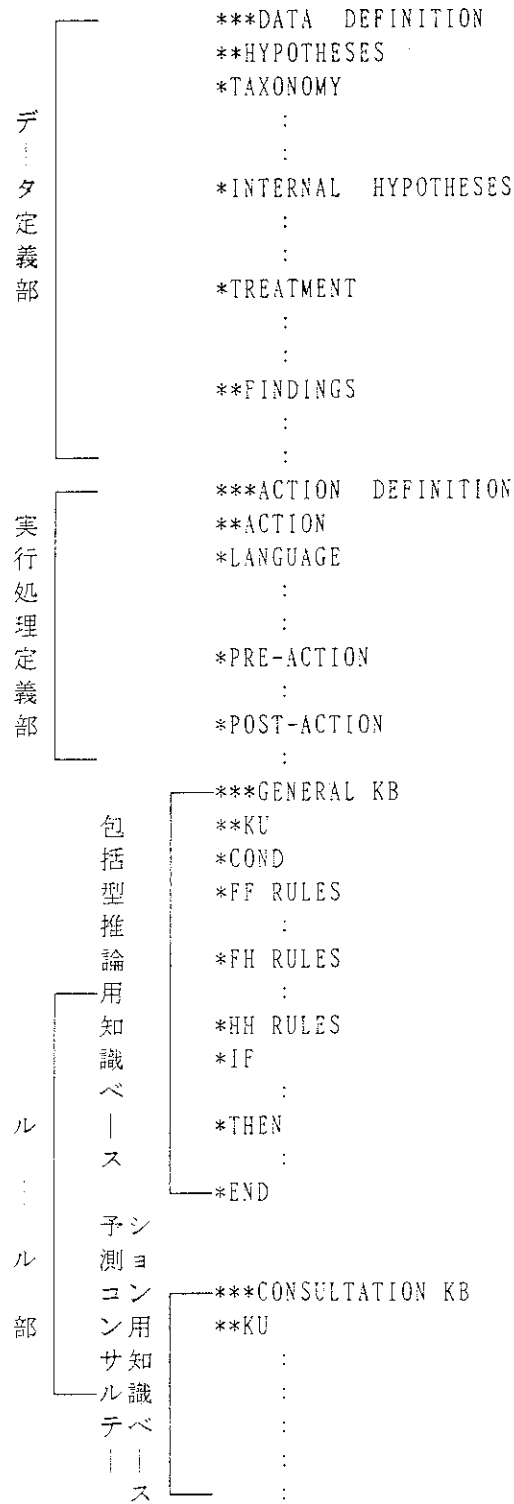


図2. 1 知識ベースの記述形式

2. 2 データ定義部

データ定義部は ***DATA DEFINITIONで定義し、仮説 (HYPOTHESES) および生起事象は、それぞれ **HYPOTHESES、**FINDINGSで定義する (図 2. 2)。

```

***DATA DEFINITION
**HYPOTHESES
:
:
:
:
**FINDINGS
:
:
:
    
```

図 2. 2 データ定義部の形式

2. 2. 1 仮説定義 (HYPOTHESES)

**HYPOTHESES以降に、仮説およびその階層構造を定義する。仮説は、仮説分類 (TAXONOMY)、内部仮説 (INTERNAL HYPOTHESES)、および対応策 (TREATMENT) に分かれ、それぞれ、*TAXONOMY、*INTERNAL HYPOTHESESおよび *TREATMENTで定義される。

仮説の階層構造とは、仮説を上位仮説、下位仮説というように階層的に定義することにより、上位仮説から下位仮説へと原因の絞り込みが系統的行える。図 2. 3は階層が3の場合の仮説分類の定義である。内部仮説、対応策についても同様に定義される。

**HYPOTHESES			
*TAXONOMY			
MNE1	TAX-DEF1	(wt1)	MNE..... : 仮説略称名
MNE11	.TAX-DEF11	(wt11)	TAX-DEF... : 仮説定義
MNE12	.TAX-DEF12	(wt12)	wt..... : 数値, $0 \leq wt \dots \leq 1$
MNE121	..TAX-DEF121	(wt121) : 階層の深さを表す。
MNE13	.TAX-DEF13	(wt13)	

図 2. 3 階層的仮説 (仮説分類) 定義例

① 仮説分類 (TAXONOMY)

仮説分類は、*TAXONOMYで定義する。図2.3で示したように階層的に定義することもできる。図2.4に仮説分類の定義形式、図2.5に具体例を示す。

```

**HYPOTHESES
*TAXONOMY
MNE          TAX-DEF
:            :
:            :
:            :

*INTERNAL HYPOTHESES
:
:
:
    
```

MNE : 仮説分類の略称名

TAX-DEF : 仮説分類の定義

図2.4 仮説分類定義形式

```

*TAXONOMY
NRA          NUCLEAR POWER PLANT ACCIDENT
FPCS         . FAILURE OF PRIMARY COOLANT SYSTEM (0.18)
FPCSPM       .. FAILURE OF PRIMARY COOLANT PUMP (0.60)
RCSPL1       ... LOSS OF REACTOR COOLANT ONE PUMP (0.30)
RCSPL2       ... LOSS OF REACTOR COOLANT TWO PUMP (0.20)
RCSPLA       ... LOSS OF REACTOR COOLANT ALL PUMP (0.20)
RCSPSL       ... LOSS OF REACTOR COOLANT ONE PUMP SHAFT LOCK (0.20)
FPCSLP       .. LEAK OF PRIMARY COOLANT LOOP (0.30)
RCSUTR       ... SG U-TUBE RUPTURE (0.25)
RCSLSL       ... SMALL LEAK OF REACTOR COOLANT SYSTEM (0.30)
RCRVSO       ... PRESSURIZER RELIEF VALVE STICKS OPEN (0.30)
    
```

図2.5 仮説分類定義例

② 内部仮説 (INTERNAL HYPOTHESES)

内部仮説は、*INTERNAL HYPOTHESESで定義する。図2.3で示したように、階層的に定義することもできる。図2.4に内部仮説の定義形式を図2.7に具体例を示す。

```

**HYPOTHESES
*TAXONOMY
:
:
*INTERNAL HYPOTHESES
MNE      IH-DEF
:         :
:         :
*TREATMENT
:
:
    
```

図2.6 内部仮説定義形式

MNE : 内部仮説の略称名
 IH-DEF : 内部仮説の定義

```

*INTERNAL HYPOTHESIS

IHTAX1  APPLY CONDITION OF KNOWLEDGE UNIT 2
IHTAX2  APPLY CONDITION OF KNOWLEDGE UNIT 3
FRADM   FAILURE OF RADIATION MONITER
    
```

図2.7 内部仮説定義例

③ 対応策 (TREATMENT)

対応策は、*TREATMENTで定義する。図2.3で示したように階層的に定義することもできる。図2.8に対応策の定義形式を図2.9に具体例を示す。

```

**HYPOTHESES
*TAXONOMY
      :
      :
*INTERNAL HYPOTHESES
      :
      :
*TREATMENT
MNE      TR-DEF
:         :
:         :
    
```

MNE : 対応措置の略称名
 TR-DEF : 対応措置の定義

図2.8 対応措置定義形式

```

*TREATMENTS

TREAT01    TAKE ACTION OF TRETMENT NO.1
TREAT02    TAKE ACTION OF TRETMENT NO.2
TREAT03    TAKE ACTION OF TRETMENT NO.3
TREAT04    TAKE ACTION OF TRETMENT NO.4
TREAT05    TAKE ACTION OF TRETMENT NO.5
TREAT06    TAKE ACTION OF TRETMENT NO.6
    
```

図2.9 対応措置定義例

2. 2. 2 生起事象定義 (FINDINGS)

生起事象は、**FINDINGS以降で定義する。生起事象は次の4つに分けて定義できる。

- ① MULTIPLE CHOICE (一価選択型事象)

*MULTIPLEにより定義する。

- ② CHECKLIST (多枝選択型事象)

*CHECKLISTにより定義する。

- ③ YES/NO (YES/NO型事象)

*YES/NOにより定義する。

- ④ NUMERICAL (数値型事象)

*NUMERICALにより定義する。

さらに、値の設定方法は、次の3つのタイプに分けて指定できる。

- (ア) Iタイプ

ルールの適用前にユーザーがケースファイルを利用して値を設定する。

- (イ) Eタイプ

ルールの適用された時点で、ユーザーに問い合わせがされ、端末から値を入力する。

- (ウ) Rタイプ

F Fルールおよび、HHルールのサブルールにより値を設定する。図2. 10にルール中でのRタイプの生起事象の使用形式を示す。

なお、省略時はEタイプとなる。

$$F(mne, T) \rightarrow F(mner, F)$$

mne : 生起事象の略称名

mner : Rタイプの生起事象の略称名

図2. 10 Rタイプの生起事象のルール中での使用形式

また、生起事象に値が与えられると生起事象の略称名には図2. 11に示されるような時間と値とを組にしたリリストを要素とするリストがバインディングされる。これを生起事象リストという。生起事象値リストの要素数の最大値は生起事象定義で指定する。

$$((V_1 T_1)(V_2 T_2) \dots (V_n T_n))$$

V_i : 生起事象値

T_i : 時刻

図2. 11 生起事象値リスト

生起事象の定義形式を図2. 12に示す。

```

**FINDINGS
*MULTIPLE CHOICE (NT1 TYPE1)
    :
    :
    :
*YESNO (NT2 TYPE2)
    :
*NUMERICAL (NT3 TYPE3)
    :
*YESNO (NT4 TYPE4)
    :
    
```

NT1,NT2,NT3,NT4 : 生起事象値リストの要素数を示す整数、省略時は1である。

TYPE1,TYPE2,TYPE3,TYPE4 : 値設定方法の指定を示す。
 Iなら、Iタイプ生起事象、
 Eなら、Eタイプ生起事象、
 Rなら、Rタイプ生起事象、
 省略時はEである。

図2. 12 生起事象の定義形式

① 一価選択型事象 (MULTIPLE CHOICE)

一価選択型事象は、*MULTIPLE CHOICEで定義する。図2.13に定義形式を示す。図2.14に具体例を示す。

```

*MULTIPLE CHOICE (NT TYPE)
HEADER :
MNE1    MC-DEF1
MNE2    MC-DEF2
:
:

```

NT : 生起事象値リストの要素数を示す整数
TYPE : 値の設定方法を示す。I、E、Rのいずれかで、省略時はEがとられる。値はTまたはFである。
HEADER : 値をユーザーに問い合わせるときに先頭に示される文字例
MNE1,MNE2 : 生起事象の略称名
MC-DEF1,MC-DEF2 : 生起事象の定義

図2.13 一価選択型事象定義形式

```

*MULTIPLE CHOICE
TYPE OF PROBLEM:
NPPA    NUCLEAR POWER PLANT ACCIDENT
OTHA    OTHER ACCIDENT

```

図2.14 一価選択型事象定義例

② 多枝選択型事象 (CHECKLIST)

多枝選択型事象は、*CHECKLIST で定義する。図 2. 15 に定義形式を、図 2. 16 に具体例を示す。

```
*CHECKLIST      (NT TYPE)
HEADER:
MNE1  CL-DEF1
MNE2  CL-DEF2
:
:
```

NT : 生起事象値リストの要素数を示す整数
TYPE : 値の設定方法を示す。I、E、Rのいずれかで、省略時はEがとられる。値はTまたはFである。
HEADER : 値をユーザーに問い合わせるときに先頭に示される文字例
MNE1,MNE2 : 生起事象の略称名
CL-DEF1,CL-DEF2 : 生起事象の定義

図 2. 15 多枝選択型事象定義形式

```
*CHECKLIST
ESS SIGNAL:
ESSACT  ESS SIGNAL ACTUATION
ESCOPH  CONTAINMENT PRESSURE HIGH
ESPRLL  PRESSURIZER BOTH LEVEL AND PRESS. LOW
ESSGHP  DIFFER. PRESS. SG AND S.H HIGH
ESSTFH  STEAM FLOW HIGH
ESSLPL  STEAM LINE PRESSURE LOW
ESTAVL  AVERAGE TEMPERATURE LOW
```

図 2. 16 多枝選択型事象定義例

③ YES/NO型事象

YES/NO型事象は、*YESNOで定義する。図2. 17に定義形式を示す。図2. 18に具体例を示す。

```
*YESNO (NT TYPE)
MNE YN-DEF
```

NT : 生起事象値リストの要素数を示す整数
 TYPE : 値の設定方法を示す。I、E、Rのいずれかで、省略時はEがとられる。値はTまたはFである。
 MNE : 生起事象の略称名
 YN-DEF : 生起事象の定義

図2. 17 YES/NO型事象定義形式

```
*YESNO (5)
YREVFL1 REVERSE FLOW LOOP 1
```

図2. 18 YES/NO型事象定義例

④ 数値型事象 (NUMERICAL)

数値型事象は、*NUMERICALで定義する。図2. 19に定義形式を示す。図2. 20に具体例を示す。

```
*NUMERICAL (MIN MAX) (NT TYPE)
MNE NU-DEF
```

- MIN : 数値または ' * '、値の下限をチェックする。
 値がMIN以下のときは、値はMINになる。
 *のときは下限チェックなし。
- MAX : 数値または ' * '、値の上限をチェックする。
 値がMAX以上のときは、値はMAXになる。
 *のときは上限チェックなし。
- NT : 生起事象値リストの要素数を示す整数。
- TYPE : 値の設定方法を示す。I、E、Rのいずれかで、
 省略時はEがとられる。
- MNE : 生起事象の略称名
- NU-DEF : 生起事象の定義

図2. 19 数値型事象定義形式

なお、値の上限、下限の両方ともチェックしないときは (MIN MAX)は省略できる。

```
*NUMERICAL (7 I)
TEMCL3 COLD LEG TEMPERATURE NO.3 LOOP (DEG.C)
```

図2. 20 数値型事象定義例

2. 3 ルール部

ルール部では包括型知識ベースおよび予測コンサルテーション型知識ベースを定義する。包括型知識ベースは、***GENERAL KBで定義され、予測コンサルテーション型知識ベースは、***CONSULTATION KBで定義される。それぞれ複数の知識ユニットで構成される。(図 2. 2 1)

```

***GENERAL KB
**KU
:
:
**KU
:
:
**KU
:
:
***CONSULTATION KB
**KU
:
:
**KU
:
:
    
```

図 2. 2 1 ルール部定義形式

2. 3. 1 知識ユニット定義

知識ユニットは、**KUで定義される。知識ユニットは、条件部、FFルール、FHルール、HHルールから構成される。(図2. 22)

```

**KU
*COND .....
*FF RULES
:
:
*FH RULES
:
:
*HH RULES
:
:
    
```

図2. 22 知識ユニット定義形式

(1) 知識ユニットの適用条件

*CONDで知識ユニットの適用条件を定義する(図2. 23)。この適用条件が推論実行に際して知識ユニットの適用可否を決定する。知識ユニットの適用条件は省略可能で、省略した場合は、その知識ユニットは無条件に適用される。

適用条件には、生起事象とその値、または仮説とその確信度およびそれらを結合したものが図2. 23のように記述される。この例では、生起事象AHPが真で仮説AGEが0.3以上0.4以下であるとき、この知識ユニットのルールが適用される。

知識ユニットの適用条件は、ルールの条件部と同じ形式で記述される。記述形式の詳細は2.3.2のルールの項で述べる。

```

**KU
*COND F(AHP,T) & H(AGE,0.3:0.4)
*FF RULES
:
:
    
```

図2. 23 知識ユニットの適用条件定義例

(2) FFルール定義

図2.24に示すように、*FF RULESでFFルールを定義する。また、*FF RULESでは、ネガティブルール、メッセージ表示ルール、ポーズルール、プログラム実行ルールも定義できる。

```

**KC
*COND
*FF RULES
F (PRCONT,0.62 : *) -> F (AINISG,T)
F (LCRODD,0 : 10) -> F (ACRDBTM,T)
:
:
*FH RULES
:
:
*HH RULES
:
:

```

図2.24 FFルールの定義例

(3) FHルール定義

図2.25に示すように、*FH RULESで、FHルールを定義する。また、*FH RULESでは、ネガティブルール、メッセージ表示ルール、ポーズルール、プログラム実行ルールも定義できる。

```

**KU
*COND
*FF RULES
      :
      :
*FH RULES
F (SPRZPL,T)  -> H (FPRZ,0.40)
F (SSGLEL,T)  -> H (FPCS,0.50)
      :
      :
*HH RULES
      :
      :

```

図2.25 FHルールの定義例

(4) HHルール定義

図2.26に示すように、*HH RULESでHHルールを定義する。個々のHHルールについては、*IFで条件部を、*THENで実行部、すなわちHHルールのサブルールを定義する。1つのHHルール定義は必ず*ENDで終わる。HHルールの条件部には、生起事象および仮説に関する条件を記述する。サブルールは、条件部に生起事象または仮説に関する条件を持ち、実行部にはFFルール、FHルールを持つ。

また、サブルールとしては、HHルール以外のすべてのルール、すなわち、FFルール、FHルール、ネガティブルール、メッセージ表示ルール、ポーズルール、プログラム実行ルールも記述できる。

```

**KU
*COND
*FF RULES
:
:
*FH RULES
:
:
*HH RULES
*IF
H (FCRD,0.1:1.0)
*THEN
H (FPCS,0.5 : 0.9) -> F (AINISI,T)
*END
*IF
:
:
*END

```

図 2. 26 HHルール定義例

2. 3. 2 ルールの定義

(1) 条件部の定義

ここでいう条件部とは、ルールの条件部・知識ユニットの適用条件・HHルールのサブルールの条件部のことである。

条件部では、ある1つの生起事象の値またはある1つの仮説の確信度についての記述を単位とし、それを単独または結合して記述する。

条件部の単位となるのは次のものである。

- ①

F (Mne,Valf)

- ②

F (Mnel,#.Valf)

- ③ $F (Mne2, Num1 : Num2)$
- ④ $F (Mne2, \#, Num1 : Num2)$
- ⑤ $H (Mne3, Cf1 : Cf2)$

①～⑤において、各記号は次のことを示す。

- Mne1, Mne2 : 生起事象の略称
- Mne3 : 仮説の略称
- Valf : CHECKLIST, MULTIPLE CHOICE, YES/NOタイプの生起事象の生起事象の値で、'T'または'F'
- Num1 : Num2 : NUMERICALタイプの生起事象の値を指定する数値または'*', $Num1 \leq Num2$ 。'*'は、無限大または無限小をあらわす。
- Cf1 : Cf2 : 仮説の確信度を指定する-1以上1以下の数値または'*', $Cf1 \leq Cf2$ 。'*'は、無限大または無限小をあらわす。
- # : その条件の、過去における成立も考慮に入れる。

以上の5つの単位を単独で条件部で記述するだけでなく、次の示すように結合してより複雑な条件を記述できる。

⑥ COUNT結合

(m: X1, X2, …… Xn)

- X1, X2, …… Xn : ①～⑤の単独条件、COUNT結合、AND結合をカッコでくくったもの、またはAFTER結合をカッコでくくったもの
- m, n : 正の整数, $0 \leq m \leq n$ を満たす
n個の条件のうち m個の条件が満たされれば、このCOUNT結合は満足される

⑦ AND結合

$$X1 \ \& \ X2 \ \& \ \cdots \ \& \ Xn$$

$X1, X2, \dots, Xn$: ①～⑤の単独条件、COUNT結合またはAFTER結合。
 n : 正の整数

n 個のすべての条件が満たされたとき、AND結合は満足される。

⑧ AFTER結合

$$X1, \text{AFTER}(\gamma_1: \gamma'_1), X2, \dots, X_{n-1}, \text{AFTER}(\gamma_{n-1}, \gamma'_{n-1}), Xn$$

$X1, X2, \dots, Xn$: ①～⑤の単独条件、COUNT結合またはAND結合をカッコでくくったもの。
 $\gamma_1, \gamma'_1, \gamma_{n-1}, \gamma'_{n-1}$: 時間を表す数、整数部は4桁までで、
 上2桁が時間、次の下2桁が分、小数点以下2桁が秒を表す(7.45; 7分45秒)。
 $\gamma_1 < \gamma'_1 < \gamma_2 < \gamma'_2 < \dots < \gamma_{n-1} < \gamma'_{n-1}$
 n : 正の整数

Xn が満たされた後、 γ_{n-1} 以後、 γ'_{n-1} 以内に X_{n-1} が満たされ、 X_{n-1} が満たされた後、 γ_{n-2} 以後、 γ'_{n-2} 以内に X_{n-2} が満たされ、……、 $X2$ が満たされた後、 γ_1 以後 γ'_1 以内に $X1$ が満たされたとき、このAFTER結合は満足される。

(2) 実行部の定義

① FFルール

条件部 : 生起事象に関する条件のみ可

実行部 : $F(Mne, FVAL)$

Mne : Rタイプの生起事象の略称名

$FVAL$: 数値または`T`または`F`

実行処理 : 生起事象 Mne に値 $FVAL$ を与えることを意味する。

② FHルール

条件部 : 生起事象に関する条件のみ可

実行部 : $H(Mne, Cf)$

Mne : 仮説の略称名

Cf : 数値, $-1 \leq Cf \leq 1$

実行処理 : 仮説Mneに確信度 Cfを割りふることを意味する。

③ メッセージ表示ルール

条件部 : 生起事象または仮説に関する条件

実行部 : $M("SENT1" \dots "SENT2" \dots \dots "SENTn")$

SENT1, SENT2, ... SENTn : 任意の文字列

n : 正の整数

実行処理 : メッセージ表示ルールが適用され、条件部が満足されると SENT1, SENT2, ... , SENTn が1行ごとに画面上に表示される。

④ ポーズルール

条件部 : 生起事象または仮説に関する条件

実行部 : $P("SENT1" \dots "SENT2" \dots \dots "SENTn")$

SENT1, SENT2, ... SENTn : 任意の文字列

n : 正の整数

実行処理 : ポーズルールが適用され、条件部が満足されると、そこで一時ルールの適用を中断し、SENT1, SENT2, ... , SENTn が1行ごとに画面上に表示した後コンサルテーションを開始する。ンサルテーションを終了すると再ルールの適用が開始される。

⑤ プログラム実行ルール

条件部 : 生起事象または仮説に関する条件

実行部 : $I(\text{PROGRAM-NAME})$

PROGRAM-NAME : 実行処理定義部で定義したリスエスト
実行タイプのアクション名

実行処理 : アクション PROGRAM-NAMEを実行する。

2. 4 実行処理定義部

実行処理定義部では、***ACTION DEFINITIONでアクションを定義する。定義形式を図2. 27に示す。

```

***ACTION DEFINITION
**ACTION ACT-NAME1
      :
      :
      :
**ACTION ACT-NAME2
      :
      :
      :
      :
      :

```

ACT-NAME1 , ACT-NAME2 : アクション名

図2. 27 実行処理定義形式

2. 4. 1 アクション定義

図2. 28に示す形式でアクションを定義する。図2. 29に具体例を示す。

```

**ACTION  ACT-NAME
*LANGUAGE LG  TYPE
PROCESS1
PROCESS2
:
:
*PRE-ACTION
PREACTION1
PREACTION2
:
:
*POST-ACTION
POSTACTION1
POSTACTION2
:
:

```

- ACT-NAME : アクション名
- LG : 使用言語名、省略時はL I S Pである。
- TYPE : アクション実行のタイミングを指定する。I
なら、すべてのルール適用開始直前、fなら、
すべてのルール適用終了直後Rまたは省略時
には、ルール適用時にアクションが実行され
る。
- PROCESS1, PROCESS2, : アクションの内容の記述である。
使用言語がL I S Pの場合はS式である。
それ以外の言語の場合は、T S Sコマンドを
記述する。T S Sコマンドのパラメータは
“ ” でかこむ。
- PREACTION1, PREACTION2 : L I S PのS式の記述である。アクション実
行直前に評価される。
- POSTACTION1, POSTACTION2 : L I S PのS式の記述である。アクション
実行直後に評価される。

図 2. 2 2 アクション定義形式

```
***ACTION DEFINITION
**ACTION INITIAL-SET1
*LANGUAGE LISP I
(DEFUN D*INITIAL-SET NIL
  (COND ((NOT(BOUNDP' /#/#KB-STATE))(D*VALUE-SET-NUM 'INITIAL-SET
                                                    '(INITIAL 0.1)))
        (T (D*VALUE-SET-NUM 'INITIAL-SET
                              (LIST 'INITIAL /#/#KB-STATE))))))
(D*INITIAL-SET)
**ACTION FINAL-SET1
*LANGUAGE LISP F
(DEFUN D*FINAL-SET NIL
  (SETQ /#/#KB-STATE (CAAAR INITIAL)))
(D*FINAL-SET)
```

図 2. 2 3 アクション定義例

3. 操作マニュアル

3.1 操作の概要

D I S K E Tの主な機能を以下に示す。

① 知識ベースの構築支援機能

決定木出力、確信度の簡易計算、ルールの説明、画面の二分割、知識ベースエディタ等により、知識ベースの構築を行う。

② コンパイル機能

ユーザーが、知識ベース構築支援機能を用いて作成、修正した知識ベース・ソースを推論実行に利用できる表現に変換する。

③ ケースデータ作成機能

推論実行に先立ち、生起事象に値および時間データを設定する機能であり、ファイルに保存するためのケースジェネレータと修正、追加用のケースエディタとがある。

④ 推論実行機能

コンパイルされた知識ベースとケースデータを用いて、前向き推論を行う。

⑤ コンサルテーション機能

前向き推論の結果について、コンパイルされた知識ベースを基に生起事象、仮説、ルール等のコンサルテーションを行う。

この他にロード、セーブ、知識ベース検索、知識ベーステスト等の機能がある。

D I S K E Tのプログラムは機能単位に分割されてファイルに保存されている。使用にあたっては関連した機能を組合わせて設定できる。この組合せは、①～⑤の主な機能にその他の機能を付加したものが中心になっている。また、この組合せのことを”モード”と呼ぶ。

D I S K E Tを実行するにはコマンドプロシジャーを実行する。このとき図3.1に示すようなモードを選択が可能である。コマンドプロシジャーが実行されるとU T I L I S Pが起動され、各モードの初期画面が表示される。ここからさらにメニューを選択し、個々の機能を使用する。各モードの詳細については、3.2節で示す。

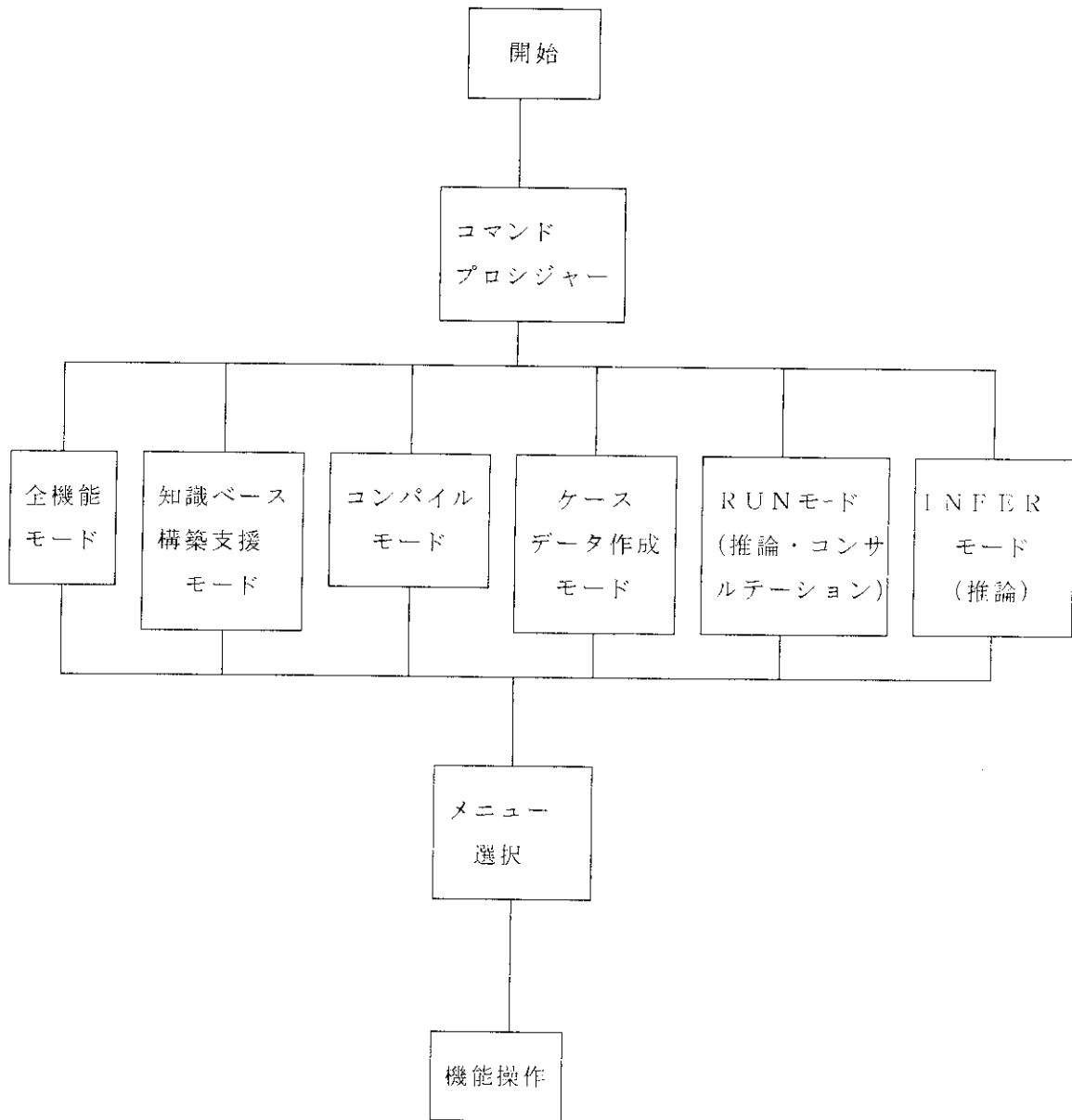


図3.1 DISKETの操作の概要

3.2 モードの選択

コマンドプロシジャーを実行することにより、モードを選択し、必要な機能を起動する。

3.2.1 起動

```
LOGON TSS... S(4000)
```

と入力する。(但しグラフィックス機能を使用するときには、D-SCAN端末においてTSSの代わりにSUNを入力する。)

3.2.2 モード選択

i) コマンドプロシジャーの実行

```
DISKET parameter (sub-parameter)
```

と入力した後、UTILISPのプロンプトが表示されるので、

```
(IERIAS)
```

と入力する。

parameterの指定によりモードを選択する。表3.1にparameterと起動される機能の対応を示す。(parameterとしてTOOLを指定した場合はUTILISPのプロンプトは表示されず、再びリターンキーを入力すればよい。)またsub-parameterは複数指定することが可能で、sub-parameterとしては表3.2に示すものがある。

parameterとしてALLを指定し且つ下位コマンドPFDを使用する場合、PROG(n) (nは200~400程度)を同時に指定しなければならない。

表 3.1 コマンドプロジジャー; DISKETのparameter

parameter	モード	起動される機能
ALL	全機能 モード	全機能
TOOL	知識ベース 構築支援 モード	知識ベース構築支援、ロード、セーブ、 コンパイル
COMPILE	コンパイル モード	コンパイル、セーブ、ヘルプ
CASE	ケースデータ 作成モード	ケースジェネレータ、ケースエディタ、ロード、 セーブ、ヘルプ
RUN	RUNモード	推論実行、コンサルテーション、ロード、 知識ベーステスト、知識ベース検索、ヘルプ
INFER	INFERモード	推論実行、ロード、ヘルプ

表 3.2 コマンドプロジジャー; DISKETのsub-parameter

sub-parameter	起動される機能
PROG(n)	n kbyteの外部プログラム用エリアの確保 (ALL(PFD), GRAPHICS を用いるとき必要)

ii) メニュー選択

各モードが起動されると初期画面が表示され、セットアップされた機能のメニューが表示されるので、これを選択する。ただし知識ベース構築支援機能の場合は初期画面で、知識ベースファイル名を入力する。(詳細は次節)

3.2.3 終了

i) 各機能の終了

各機能のメニュー選択で

Q U I T

と入力するとDISKETが終了しUTILISPのプロンプトが表示される(知識ベース構築支援機能の場合はファンクションキーPF3を押すことによる)。

ii) UTILISPの終了

UTILISPのプロンプトが表示されている状態で、

(Q U I T)

と入力すると、UTILISPが終了し、TSSのREADY状態になる。

iii) TSSの終了

L O G O F F

と入力する。

以下にDISKETの起動から終了までの入力例を示す。

```

READY
DISKET RUN GRAPHICS PROG(200)
> (IERIAS)
# *****
# *
# *   WELCOME TO IERIAS   (VERSION 3. ) *
# *   EXECUTION MODE    *
# *   1986-03           *
# *
# *
# *****

# ENTER COMMAND <LOAD/RUN/LIST/TEST/HELP/COMMAND/QUIT>=> LOAD
# ENTER FILE NAME=> "J2749.SRSNC.DATA"

# *****  LOADING END  *****
:
:
:
:

# ENTER COMMAND <LOAD/RUN/LIST/TEST/HELP/COMMAND/QUIT>=> QUIT
# *****  PROCESSING END  *****
NIL
> (QUIT)
KEQ57023I TESTED PROGRAM TERMINATED NORMALLY+
READY

```

図3.2 DISKETの起動と終了

3.3 モードの操作

ここではコマンドプロシジャ-の実行により表3.1に示される機能が起動された後の操作を表3.1の分類に従って説明する。

3.3.1 全機能モード

図3.3に示されるメッセージが表示されるのでメニューを入力する。

```
# *****
# *
# * WELCOME TO IERIAS (VERSION 3. ) *
# *
# * 1986-03 *
# *
# *****

# ENTER COMMAND <COMP/ED/LOAD/SAVE/CASE/RUN/LIST/PFD/TEST/TOOL/QUIT>=>
```

図3.3 初期画面（全機能）

メニューと機能の対応を表3.3に示す。

表3.3 全機能モードでのメニューと下位機能

メニュー	下位機能
COMP	コンパイル
ED	エディタ
LOAD	ロード
SAVE	セーブ
CASE	ケースジェネレータ
RUN	推論実行
LIST	知識ベースの検索
PFD	PFD
TEST	知識ベースのテスト
TOOL	知識ベース構築支援
QUIT	終了

3.3.2 知識ベース構築支援モード

図3.4に示されるメッセージが表示されるので、知識ベースの保存されているファイル名を入力する。コンパイルされた知識ベース名が入力されれば、知識ベース構築支援画面に移り、メニューが入力できる。知識ベース・ソース名を入力すると直接知識ベースエディタが呼ばれる。なお入力欄は記号 => の右側である。終了はファンクションキーPF3で行う。

```

DISKET KNOWLEDGE-BASE DEVELOPMENT TOOL

      KNOWLEDGE-BASE FILE =>

      COMPILED KD FILE    =>

      PF3 KEY ==> QUIT

```

図3.4 知識ベース構築支援モード初期画面

3.3.3 コンパイルモード

図3.5に示されるメッセージが表示されるのでメニューを入力する。
(各メニューと下位機能の関係は、表3.3を参照のこと。)

```

# *****
# *
# *   WELCOME TO IERIAS   (VERSION 3. ) *
# *   COMPILER MODE      *
# *   1986-03            *
# *
# *****

# ENTER COMMAND <COMP/SAVE/HELP/QUIT>=>

```

図3.5 コンパイルモード

3.3.4 RUNモード

図3.6に示されるメッセージが表示されるのでメニューを入力する。
 (各メニューと下位機能との関係は、表3.3を参照のこと。)

```
# *****
# *
# *   WELCOME TO IERIAS   (VERSION 3. ) *
# *   EXECUTION MODE     *
# *       1986-03        *
# *
# *****

# ENTER COMMAND <LOAD/RUN/LIST/TEST/HELP/COMMAND/QUIT>=>
```

図3.6 RUNモード

3.3.5 INFERモード

図3.7に示されるメッセージが表示されるのでメニューを入力する。
 (各メニューと下位機能との関係は、表3.3を参照のこと)

```
# *****
# *
# *   WELCOME TO IERIAS   (VERSION 3. ) *
# *   ONLY EXECUTION MODE *
# *       1986-03        *
# *
# *****

# ENTER COMMAND <LOAD/RUN/QUIT/HELP>=>
```

図3.7 INFERモード

3.4 各機能の操作

3.4.1 知識ベース構築支援

i) 開始

知識ベース構築支援機能が起動されると、図3.8に示す画面が表示される。コンパイルされた知識ベースの保存されているファイル名を入力すると、その知識ベースがローディングされ図3.9に示す画面が表示される。この画面の第1行右側が知識ベース構築支援メニューの入力欄である。第2行以下にはメニューの実行結果が表示される。知識ベース構築支援メニュー欄を表3.4に示す。

また初期画面で、KNOWLEDGE-BASE FILE => の欄に知識ベース・ソースの保存ファイル名の入力があるとそのファイルは知識ベース・エディタで用いられ、画面は知識ベース・エディタ画面になる（図3.10）。

次節以降で主な知識ベース構築支援機能について説明する。

```
DISKET KNOWLEDGE-BASE DEVELOPMENT TOOL

KNOWLEDGE-BASE FILE =>
COMPILED KB FILE    =>

PF3 KEY ==> QUIT
```

図3.8 知識ベース構築支援初期画面

表 3.4 知識ベース構築支援メニュー一覧

メニュー (カッコ内は省略可)	機 能
CANCEL	知識ベース・ソースを保存せずに終了
END	知識ベース・ソースを保存して終了
SETH 仮説名	対象仮説の設定
SAVE ("ファイル名")	知識ベース・ソースの保存
LOAD "ファイル名"	知識ベース・ソースのロード
SAVEC "ファイル名"	コンパイルされた知識ベースの保存
LOADC "ファイル名"	コンパイルされた知識ベースのロード
LOADCASE "ファイル名"	ケースデータのロード
DTREE	決定木出力
SUBTREE (node)(level)	部分木出力
SETC (生起事象名)	生起事象値の設定
CALC TIME-ID	確信度の簡易計算
CLEARF (生起事象名)	生起事象値のクリア
RULES	関連ルール表示
FINDING	関連生起事象表示
RULEEXP ルール名	ルールの説明
COMPILE ("ファイル名")	知識ベースのコンパイル
DIV	画面二分割
UNDIV	画面二分割解除
ED	知識ベース・エディタ

```

*ENVIRONMENT* =                ==>

LOAD NORMAL ENDED

```

図 3.9 知識ベース構築支援画面

```

DISKET-EDITOR 'J2749.SRSN.DATA'      COMMAND ==>
00000 =====TOP OF KB =====
00010 #
00020 # KNOWLEDGE BASE OF SRS (OCT. 1983)
00030 #
00040 ***DATA DEFINITION
00050 **HYPOTHESES
00060 *TAXONOMY
00070 NRA      NUCLEAR POWER PLANT ACCIDENT
00080 FPCS      .FAILURE OF PRIMARY COOLANT SYSTEM      (0.18)
00090 FPCSPM    ..FAILURE OF PRIMARY COOLANT PUMP      (0.60)
00100 RCSPL1    ...LOSS OF REACTOR COOLANT ONE PUMP    (0.30)
00110 RCSPL2    ...LOSS OF REACTOR COOLANT TWO PUMPS  (0.20)
00120 RCSPLA    ...LOSS OF REACTOR COOLANT ALL PUMPS  (0.20)
00130 RCSPSL    ...LOSS OF REACTOR COOLANT ONE PUMP SHAFT LOCK (0.20)
00140 FPCSLP    ..LEAK OF PRIMARY COOLANT LOOP (0.30)
00150 RCSUTR    ...SG U-TUBE RUPTURE (0.25)
00160 RCSLSL    ...SMALL LEAK OF REACTOR COOLANT SYSTEM (0.30)
00170 RCRVSO    ...PRESSURIZER RELIEF VALVE STICKS OPEN (0.30)
00180 #
00190 FORZ      .FAILURE OF PRESSURIZER SYSTEM (0.12)
00200 FPRPRC    ..FAILURE OF PRESSURIZER PRESSURE CONTROL SYSTEM (0.50)
00210 PSPVO1    ...PRESSURIZER SPARY CONTROL ONE VALVE FAILS OPEN (0.20)
00220 PSPVC2    ...PRESSURIZER SPARY CONTROL TWO VALVES FAIL CLOSE (0.20)

```

図 3.10 知識ベース・エディタ画面

ii) 終了

図 3.9 および図 3.10 の画面でコマンド CANCEL あるいは END を入力すると、図 3.8 の初期画面に戻り、さらにファンクションキー PF3 を押すと、知識ベース構築支援機能は終了する。

iii) 画面の移動

知識ベース構築支援機能では、画面出力のためのバッファを持つので任意の部分を画面をスクロールして表示することができる。スクロールはファンクションキーにより行われ、PF7～PF10を使用する。

- ① PF 7 上へページスクロール
- ② PF 8 下へページスクロール
- ③ PF 9 カーソル位置より上へスクロール
- ④ PF 10 カーソル位置より下へスクロール

また、知識ベースエディタではLOCATEコマンドにより行単位のスクロールが行える。

iv) 決定木出力

メニューDTREEを入力すると、SETHにより設定された仮説について決定木を出力できる。図3.11は仮説FPCSの決定木である。

なお、図3.9の画面から決定木が表示されるまでの入力例を図3.12に示す。

```

*ENVIRONMENT* = FPCS          ==>
FPCS(CF=NIL) -- <+> -- FH-2(0.50) - - COUNT 1 -----SRCSL1(TRUE)
|
|                               | -SRCSL2(TRUE)
| -FH1-4(0.50) - - COUNT 1 -----FLRCS1(( 50 50))
|
|                               | -FLRCS2(( 50 50))
|
|                               | -FLRCS3(( 50 50))
| -FH1-5(0.45) - - APRECON(TRUE)
| -FH1-7(0.45) - - ARADMAL(TRUE)
| -FH1-8(0.40) - - ESSACT(TRUE)
| -FH1-10(0.40)- - COUNT 1 -----PRCONT((0.65 *))
|
|                               | -TECONT((50 *))
| -FH1-6(0.35) - - AFLCHAH(TRUE)
| -FH1-3(0.30) - - AFLRCSL(TRUE)
| -FH1-9(0.30) - - &-----ALEVPRL(TRUE)
|
|                               | -APREPR1(TRUE)
| -FH1-11(0.15)- - COUNT 1 -----ATEPRLH(TRUE)
|
|                               | ATEPRLH(TRUE)
|
|                               | -APRERLH(TRUE)
| -FH1-1(0.05) - - SPRZPL(TRUE)
    
```

図3.11 決定木出力

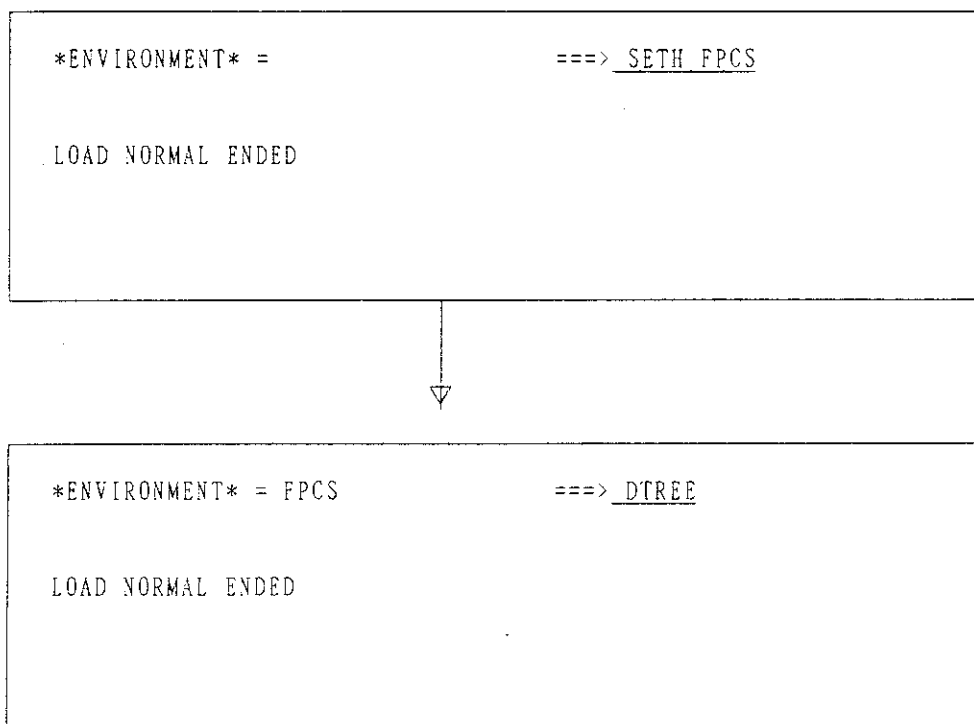


図 3.1 2 決定木出力の入力例

また部分木の表示を次に示すメニュー SUBTREE で行うことができる。

```
SUBTREE NODE LEVEL
```

部分木はNODEとLEVELの指定ができる。NODEとは、仮説や生起事象の略称名またはルール名である。LEVELは決定木または部分木の先頭のNODEから何番目までを表示するかを示す整数である。NODEの省略時は現在先頭のNODEがそのまま使われる。LEVELの省略時は、木の最も深い部分まで表示される。

v) 確信度の簡易計算

メニュー S E T H によって指定された仮説の確信度を計算する。これを実行するために、必要な生起事象に値を設定しルールを適用する。

生起事象の値の設定はメニュー S E T C で行う。S E T C を入力すると関連する生起事象が画面上に表示され、値と時間を入力できる (図 3.13)。なお生起事象値リストの要素数が 2 以上に指定されているときは、入力欄はその分表示される。

```

*ENVIRONMENT* = FPCS          ==>
  SPRZPL (PRESSURIZER LOW PRESSURE) :      =>V= T=
  SRCSL1 (RCS LOW FLOW 13) :                =>V= T=
  SRCSL2 (RCS LOW FLOW 23) :                =>V= T=
  AFLRCSL (LOW REACTOR COOLANT FLOW) :      =>V= T=
                                           =>V= T=
                                           =>V= T=
                                           =>V= T=
                                           =>V= T=
                                           =>V= T=
  FLRCS1 (REACTOR COOLANT FLOW NO.1 LOOP (%)) :  =>V= T=
                                           =>V= T=
    
```

図 3.13 生起事象値の設定

ケースデータに基づき生起事象の値の設定を行うには、メニュー L O A D C A S E によりケースファイルをローディングする (次の様に入力する)。

```
LOADCASE " ' J 2 7 4 9 . R C S U T R . D A T A ' "
```

このとき、メニュー S E T C を実行すると図 3.13 の入力欄にはケースデータによる値および時間が表示される。

メニュー S E T C で、生起事象名を指定しないときは、メニュー S E T H で指定された仮説に関連した生起事象がすべて表示される。

以上の方法で設定された生起事象値をもとに、メニュー C A L C を選択実行すれば、確信度が計算される。対象となる仮説が上位仮説を持てば上位仮説の確信度も計算される。

例えば図 3.13 の画面において

```
CALC 810
```

と入力すると（8:10における確信度の計算）、次の様に計算結果（仮説FPCSの確信度=0.89）が表示される。

```
*ENVIRONMENT* = FPCS          ==>
FPCS : 0.89
```

また、一度設定した生起事象の値をクリアするには、メニューCLEARFを選択実行する。

vi) 知識ベースエディタ

① 知識ベース表示画面

知識ベース表示、画面の1行目をコマンド入力行、2行目から23行目までを知識ベース表示行とよぶ（図3.14）。

(7) コマンド入力

コマンド入力には、行左端にタイトル、中央にファイル名が表示される。右側はコマンド入力欄である。

(i) 知識ベース表示行

知識ベースの先頭には、次のメッセージが挿入され知識ベースが表示される。

```
====TOP  OB  KB====
```

知識ベースの1行が知識ベース表示行1行（72コラム）を越えるときは画面に表示されない。左端の5桁の数字は行番号で、自動的に割りふられる。

```

DISKET-EDITOR 'J2749.SRSN.DATA'          COMMAND ==>

00000 =====TOP OF KB =====
00010 #
00020 # KNOWLEDGE BASE OF SRS (OCT. 1983)
00030 #
00040 ***DATA DEFINITION
00050 **HYPOTHESES
00060 *TAXONOMY
00070 NRA      NUCLEAR POWER PLANT ACCIDENT
00080 FPCS      . FAILURE OF PRIMARY COOLANT SYSTEM (0.18)
00090 FPCSPM    .. FAILURE OF PRIMARY COOLANT PUMP (0.60)
00100 RCSPL1   ... LOSS OF REACTOR COOLANT ONE PUMP (0.30)
00110 RCSPL2   ... LOSS OF REACTOR COOLANT TWO PUMPS (0.20)
00120 RCSPLA   ... LOSS OF REACTOR COOLANT ALL PUMPS (0.20)
00130 RCSPSL   ... LOSS OF REACTOR COOLANT ONE PUMP SHAFT LOCK (0.20)
00140 FPCSLP   .. LEAK OF PRIMARY COOLANT LOOP (0.30)
00150 RCSUTR   ... SG U-TUBE RUPTURE (0.25)
00160 RCSLSL   ... SMALL LEAK OF REACTOR COOLANT SYSTEM (0.30)
00170 RCRVSO   ... PRESSURIZER RELIEF VALVE STICKS OPEN (0.30)
00180 #
00190 FPRZ      . FAILURE OF PRESSURIZER SYSTEM (0.12)
00200 FPRPRC    .. FAILURE OF PRESSURIZER PRESSURE CONTROL SYSTEM (0.50)
00210 PSPV01   ... PRESSURIZER SPRAY CONTROL ONE VALVE FAILS OPEN (0.20)
00220 PSPVC2   ... PRESSURIZER SPARY CONTROL TWO VALVES FAIL CLOSE (0.20)

```

図 3.14 知識ベース表示画面

② 編集可能なファイル

知識ベースエディタを用いて編集しようとするファイルは、次の条件を満足しなければならない。

(ア) 既にD A S D上に存在するファイルであること。

ファイルの内容は空白であってもよいが、編集・レコード形式・レコード長・ブロック長などの情報は設定されていなければならない。

現在の区分編成ファイルに、新たなメンバーを作成することは可能である。

(イ) 編成

順編成 (P S)、あるいは区分型順編成 (P O) ファイルであること。

(ウ) レコード形式

固定長 (f) ・ブロック化 (b) であること。

(エ) レコード長

80バイト

③ コマンド

知識ベースエディタ表示画面の1行目のコマンド入力欄に入力するコマンドの一覧を示す。(表3.5)

表3.5 知識ベースエディタコマンド

コマンド	省略形	オペランド	機能
CANCEL	CAN,CNCL	なし	編集内容を保存せずに終了。
COMPILE	CO,COMP	filename	filenameで指定した知識ベースをコンパイルする。オペランド省略時は知識ベースエディタ上の知識ベースがコンパイルされる。
END	E	なし	編集内容を保存して終了。
LOAD	LD	filename	知識ベースをローディングする。
LOCATE	L,LOC	n	知識ベースをn行目より表示する。
SAVE	S,SV,SAV	filename	知識ベースをすでに登録されているファイルに保存する。
TOOL	TL		知識ベース、エディタを終了し、知識ベース構築支援モードに移る
SAVEL	SL,SVL	filename	コンパイルされた知識ベースを保存する。

④ 行コマンド

知識ベースエディタは行の挿入・削除・複写・繰り返しを行コマンドで行なう。行コマンドの一覧表を表3.4.3に示す。

表3.6 行コマンド

行コマンド	機能
A (AFTER)	複写行をこの行の後に挿入する。
B (BEFORE)	複写行をこの行の前に挿入する。
C (COPY)	行を複写する。
CC	複数行を複写する。
I (INSERT)	行を挿入する。
D (DELETE)	行を削除する。
DD	複数行を削除する。
R (REPEAT)	行を繰り返す。

(7) 行コマンド入力

行コマンドは、行番号欄に入力する。行番号欄の先行する数字、空白は無視される。

(i) 行コマンドの対

C-Aのような行コマンドの対は画面をまたがってはいけない。

(u) 行コマンドの拡張

I, Rは行コマンドの後に数字を続けることができる。その他のコマンドは無効。

⑤ 画面の移動

表示画面の移動は次の方法で行う。

(7) ファンクションキーによる移動

PF7	上へページスクロール
PF8	下へページスクロール
PF9	カーソル位置より上へスクロール
PF10	カーソル位置より下へスクロール

(i) LOCATEコマンドによる移動

知識ベース表示画面1行目のコマンド入力欄にLOCATEコマンドを入力することにより、任意の位置に画面を移動できる。位置は行番号より指定する。

vii) 画面二分割

メニューD I Vにより画面二分割が行える。下の画面はメニューS E T Cによる生起事象への値設定に使われる。上の画面はその他の知識ベース構築支援メニューの実行結果が表示される。

図3.15では、上画面に仮説F P C Sに関する決定Aが、下画面に仮説F P C Sの関連事象が表示されている。

画面の移動は上画面または下画面ごとに行われる。ファンクションキーを押すときカーソルのある方の画面が移動の対象となる。

```

*ENVIRONMENT* = FPCS          ==>
FPCC(CF=NIL) -- <+> -- FH1-2(0.50) --- COUNT 1 ----- SRCSL1(TRUE)
      1                                1- SRCSL2(RRUE)
      1- FH1-4(0.50) --- COUNT 1 ----- FLRCS1((-50 50))
      1                                1- FLRCS2((-50 50))
      1                                1- FLRCS3((-50 50))
      1- FH1-5(0.45) --- APRECON(TRUE)
      1  FH1-7(0.45) --- ARADMAL(TRUE)
      1- FH1-8(0.40) --- ESSACT(TRUE)
      1  FH1-10(0.40) -- COUNT 1 ----- PRCONT((0.65 *))
                                          1- TECONT((50 *))
      1- FH1-6(0.35) --- AFLCHAH(TRUE)
*****
SPRZPL (PRESSURIZER LOW PRESSURE) :          => V=          T=
SRCSL1 (RCS LOW FLOW 13) :                => V=          T=
SRCSL2 (RCS LOW FLOW 23) :                => V=          T=
AFLRCSL (LOW REACTOR COOLANT FLOW) :       => V=          T=
      :                                     :
      :                                     :
      :                                     :

```

図3.15 画面二分割

viii) その他の知識ベース構築支援機能

① ファイルのローディング機能

コンパイルされた知識ベース、知識ベースエディタで用いる知識ベース・ソース、およびケースデータのローディングができる。対応するメニューはそれぞれ、LOADC, LOAD, LOADCASEである。

② セーブ機能

知識ベース・エディタ上の知識ベースおよびコンパイルされた知識ベースをファイルに保存できる。対応するメニューはそれぞれ、SAVE, SAVECである。

③ 関連ルール表示機能

メニューSETH選択時に設定された仮説について、関連するルールを表示する。メニューRULESで行う。実行例を図3.16に示す。

```

*ENVIRONMENT* = FPCS          ==>
FH1-2    F(SPRZPL,T) -> H(FPCS,0.05)
FH1-3    (1:F(SRCSL1,T),F(SRCSL2,T)) -> H(FPCS,0.50)
FH1-4    F(AFLRCSL,T) -> H(FPCS,0.30)
FH1-5    (1:F(FLRCS1,-50:50),F(FLRCS2,-50:50),F(FLRCS3,-50:50)) -> H(FPCS,
0.50)
FH1-6    F(APRECON,T) -> H(FPCS,0.45)
FH1-7    (1:F(AFLCHAH,T),F(FLCHAG,20:*)) -> H(FPCS,0.35)
:
:
:
    
```

図3.16 関連ルール表示機能の実行例

④ ルール説明機能

メニューEXPRULEにより、指定したルールを説明する。
例えば図3.16の画面において

```

RULEEXP FH1-2
    
```

と入力すると、次の様な結果が表示される。

```

*ENVIRONMENT* = FPCS          ==>
FH1-2      F(SPRZPL,T) -> H(FPCS,0.05)

          F(PRESSURIZER LOW PRESSURE IS TRUE)
          -> H(FAILURE OF PRIMARY COOLANT SYSTEM IS WITH CF=0.05)

```

図 3.17 ルール説明機能の実行例

⑤ コンパイル

メニュー COMP I L Eにより、コンパイルがよばれ、コンパイルを実行する。

⑥ 関連事象の表示

メニュー S E T H選択時に設定された仮説について、関連した生起事象を表示する。メニュー F I N D I N G Sによる。実行例を図 3.18 に示す。

```

*ENVIRONMENT* = FPCS          ==>
SPRZPL      : PRESSURIZER LOW PRESSURE
SRCSL1      : RCS LOW FLOW 13
SRCSL2      : RCS LOW FLOW 23
AFLRCSL     : LOW REACTOR COOLANT FLOW
FLRCS1      : REACTOR COOLANT FLOW NO.1 LOOP (%)
FLRCS2      : REACTOR COOLANT FLOW NO.2 LOOP (%)
FLRCS3      : REACTOR COOLANT FLOW NO.3 LOOP (%)
APRECON     : CONTAINMENT PARTIAL PRESSURE HIGH
:           :
:           :
:           :

```

図 3.18 関連事象表示機能の実行例

3.4.2 コンパイル

i) 開始

DISKETのトップレベルのメニュー選択で、COMPを入力する。次に示すようにファイル名の入力促される。

```
# ENTER TEXT FILE NAME/CAN :
```

ii) ファイル名入力

コンパイルする知識ベース・ソースファイルが保存されているファイル名を入力する。ファイル名は記号" (ダブルクォート) で囲む。

iii) メッセージ表示

コンパイル開始、実行、終了、ウォーニングおよびエラーメッセージが表示される。図3.19に実行例を示す。

```
# ENTER COMMAND <COMP/ED/LOAD/SAVE/CASE/RUN/LIST/PFD/TEST/TOOL/QUIT>=>_COMP
# ENTER TEXT FILE NAME/CAN : "IER.DEV.KB(SRSKBNEW)"
# ***** COMPILE START *****
# ***** HYPOTHESES COMPILE START *****
# ***** FINDINGS COMPILE START *****
# ***** ACTION DEFINITION COMPILE START *****
# ***** ACTION DEFINITION COMPILE END *****
# ***** FORMAT DEFINITION COMPILE START *****
# ***** FORMAT DEFINITION COMPILE END *****
***
# ***** GENERALK KNOWLEDGE BASE COMPILE START *****
# UNIT CONDITION NOT FOUND GENERATE 1.

THIS KNOWLEDGE UNIT HAS NOT FF-RULE => KU2

THIS KNOWLEDGE UNIT HAS NOT FH-RULES => KU2
:
:
:
```

図3.19 コンパイル実行例

3.4.3 ケースジェネレータ

i) 開始

DISKETのトップレベルのメニュー選択でCASEを入力する。ケース名の入力
が促される。

ii) ケース名入力

ケース名とケースタイプを入力する。ケースタイプには次の2つがある。

ALL : Find-TypeがRの生起事象を除く全ての生起事象について
ケースデータのを行う。

I : Find-TypeがIである生起事象に対してのみケースデータの
入力を行う。

iii) ケースデータの作成

① checklistタイプ

checklistタイプの質問が表示されるので、項目番号と値とその値になった時刻の組を
入力する。(図3.20)

```

3. ANNUNCIATORS OF CONTROL ROD
(1) ROD STOP
(2) ROD BOTTOM
(3) ROD WITHDRAW LIMIT
CHECKLIST(2)
*=> 3 (T 830) (F 800)
*=> Q

```

図3.20 checklistのケースデータ作成

上例では、ROD WITHDRAW LIMIT が 8:00 に False、8:30 に True になった事を示す。
当該 checklist 項目に対するケースデータのを終了する時はQをする。

② multiple choiceタイプ

multiple choiceに対しては選択値の項目番号とその時の時刻の組をする。操作はC
HECKLISTとほぼ同じだが選択できる生起事象は1つだけである。(図3.21)

```

*=> (1 830) (3 800)

```

図3.21 multiple choiceのケースデータ作成

③ numericalタイプ

numericalタイプに対しては、数値とその値をとった時刻の組を入力する。

```
18. PRESSURIZER LEVEL (%)
NUMERICAL*(0) (66 830)
```

図 3.2.2 numerical のケースデータ作成

図 3.2.2 では、8時30分に66%であった事を示す。

④ Yes/Noタイプ

Yes/Noタイプに対しては、値 (T or F or U or Y or N) と、その値を取った時刻の組を入力する。(図 3.2.3)

```
69. COLD LEG TEMPERATURE HIGHER THAN HOT LEG LOOP 1
Y/N*(5)=> (Y 826)(F 820)(Y 818)
```

図 3.2.3 Yes/Noのケースデータ作成

上例では、8:18に真、8:20に偽、8:26に真となった事を示す。図 3.2.4 に実行例を示す。


```

# ENTER COMMAND <COMP/ED/LOAD/SAVE/CASE/RUN/LIST/PFD/TEST/TOOL/QUIT>=> CASE

# ***** CASE GENERATION MODE START *****

ENTER CASE NAME => CASE22
ENTER CASE TYPE (ALL/I) => ALL

1. TYPE OF PROBLEM
  (1) NUCLEARE POWER PLANT ACCIDENT
  (2) OTHER ACCIDENT
CHECKLIST(0)
*=> 1 (T 800)
*=> Q

2. ANNUNCIATORS OF TRIP
  (1) REACTOR TRIP
  (2) TURBINE TRIP
  (3) GENERATOR TRIP
CHECKLIST(0)
*=> 2 (T 815)
*=> Q

3. ANNUNCIATORS OF CONTROL ROD
  (1) ROD STOP
  (2) ROD BOTTOM
  (3) ROD WITHDRAW LIMIT
CHECKLIST(2)
*=> 3 (T 830) (F 800)
*=> Q

```

図 3. 2 4 ケースジェネレータ実行例

3. 4. 4 リストエディタ

i) メニュー入力

DISKETのトップレベルメニューでEDを選択すると、図 3. 2 5 に示すようなメッセージが表示されるので、コマンドを入力する。なお各コマンドの機能は表 3. 7 の通りである。

```
PLEASE ENTER EDIT COMMAND (M/K/KU/C/HELP).
IF YOU ENTER Q, EDIT SESSION END.
E>
```

図 3. 2 5 エディタの初期表示

表 3. 7 エディタコマンド

コマンド	機能
M	データ定義の編集 (TAXONOMY, TREATMENT, INTER-HYPO, FINDINGS)
K	知識ベースの編集 (知識ユニット単位の編集)
K U	知識ユニットの編集 (UNIT-CONDITION, FF-RULE, FH-RULE, HH-RULE)
C	ケースデータの編集 (ケースエディタ)
Q	終了

ii) データ定義の編集

エディタコマンドMを入力すると、以下の様なメッセージが表示されるので、下位コマンドを入力する。各コマンドの機能は表 3. 8 の通りである。

```
# ENTER DATA EDIT COMMAND <I/M/D/L/K/Q/HELP>=>
```

表 3. 8 データ定義編集コマンド

コマンド	機能
I	データ定義の挿入 (TAXONOMY, TREATMENT, INTER-HYPO, FINDINGS)
M	データ定義の変更 (TAXONOMY, TREATMENT, INTER-HYPO, FINDINGS)
D	データ定義の削除 (TAXONOMY, TREATMENT, INTER-HYPO, FINDINGS)
L	データ定義内容の表示 (DATA DEFINITION, KNOWLEDGE BASE)
K	キャンセル
Q	終了

① データ定義の挿入

データ定義編集コマンドでIを入力すると以下の様なメッセージが表示されるので、表 3. 9 に従って入力を行う。

```
# ENTER DATA INSERT OPTIONS =>
```

表 3.9 データ定義挿入コマンドオプション

NO	入力形式	機能
1	*X MNE1 MNE2 NAME (WT)	事故仮説分類の挿入
2	*I MNE1 MNE2 NAME (WT)	内部仮説の挿入
3	*R MNE1 MNE2 NAME (WT)	対応措置の挿入
4	*M HEADER (NT F-TYPE)	生起事象 (MULTIPLE CHOICE TYPE) の挿入
5	*C HEADER (NT F-TYPE)	生起事象 (CHECKLIST TYPE) の挿入
6	*Y MNE NAME (NT F-TYPE)	生起事象 (YES-NO TYPE) の挿入
7	*N MNE NAME (MIN MAX)	生起事象 (NUMERICAL) の挿入
パラメータ説明		
MNE1	MNE2に対する上位仮説略称名	
MNE2	挿入する仮説略称名	
WT	重み係数	
NT	生起事象値リストの要素数	
NAME	仮説または生起事象の名称	
HEADER	生起事象 (CHECKLIST, MULTIPLE-CHOICE) の先頭文字列	
F-TYPE	生起事象のタイプ (I / E / R)	
MIN	数値データの下限值 (下限値のない場合は' * ')	
MAX	数値データの上限值 (上限値のない場合は' * ')	

② データ定義の変更

データ定義編集コマンドでMを入力すると以下の様なメッセージが表示されるので、表 3.10 に従って入力を行う。

```
# ENTER MNE & OPTIONS =>
```

表 3.10 データ定義変更コマンドオプション

NO	入力形式	機能
1	MNE1 WT = WEIGHT	重み係数の変更
2	MNE1 MNE = MNE2	略称名の変更
3	MNE1 NAME = NAME1	名称の変更
4	MNE1 NT = NT1	生起事象値要素数の変更
5	MNE1 FIND = F-TYPE1	生起事象タイプの変更 (I / E / R)
6	MNE1 LIMIT = (MIN1 MAX1)	限界値の変更
パラメータ説明		
MNE1	被変更仮説略称名	
MNE2	新仮説略称名	
WEIGHT	重み係数	
NAME1	仮説または生起事象の名称	
NT1	生起事象値リストの要素数	
F-TYPE1	生起事象のタイプ (I / E / R)	
MIN1	数値データの下限值 (下限値のない場合は ' * ')	
MAX1	数値データの上限值 (上限値のない場合は ' * ')	

なお、WT, MNE, NAME, NT, FIND, LIMIT, = は固定である。

③ データ定義の削除

データ定義編集コマンドでDを入力すると以下の様なメッセージが表示されるので、任意の文字を入力する。(終了する場合はQを入力)

```

NOT ENTERED DELETE MNEMONIC

IF YOU END DATA DELETE MODE, ENTER Q.
IF NOT, ENTER ANY LETTER.

==>

```

続いて以下の様なメッセージが表示されるので、削除するデータ定義の略称名を入力する。

```
PLEASE ENTER MNEMONIC =>
```

④ データ定義内容の表示

データ定義編集コマンドでLを入力すると以下の様なメッセージが表示されるので、メニューを入力する。

```
# ENTER LIST COMMAND<HYPO/TAX/IH/TRT/FIND/KB/KU/ALL/RULE>=>
```

なおメニューの詳細は3.4.11 知識ベースの検索を参照されたい。

⑤ データ定義編集メニューのキャンセル

データ定義編集コマンドでKを入力するとデータ定義編集メニューがキャンセルされ、上位メニューに戻る。

⑥ データ定義編集メニューの終了

データ定義編集コマンドでQを入力するとデータ定義編集メニューが終了し、上位メニューに戻る。

iii) 知識ベースの編集

エディタコマンドKを入力すると、以下の様なメッセージが表示されるので、下位コマンドを入力する。各コマンドの使用法を以下に述べる。

```
KNOWLEDGE-BASE EDITOR STARTS.  
# ENTER KB EDIT COMMAND <I/D/L/LA/Q/K/HELP>=>
```

① 知識ユニットの挿入

知識ベース編集コマンドでIを入力すると、以下の様なメッセージが表示されるので、挿入する知識ユニット名を入力する。

```
ENTER KNOWLEDGE-UNIT NAME / END =>
```

② 知識ユニットの削除

知識ベース編集コマンドでDを入力すると、以下の様なメッセージが表示されるので、削除する知識ユニット名を入力する。

```
ENTER KNOWLEDGE-UNIT NAME / END =>
```

③ 知識ベース内容の表示

知識ベース編集コマンドでLを入力すると以下の様なメッセージが表示されるので、メニューを入力する。

```
# ENTER LIST COMMAND<HYPO/TAX/IH/TRT/FIND/KB/KU/ALL/RULE>=>
```

④ 知識ユニット名の表示

知識ベース編集コマンドでL Aを入力すると、登録されている知識ユニット名が表示される。

iv) 知識ユニットの編集

エディタコマンドKUを入力すると、以下の様なメッセージが表示され、表3.11に示す下位コマンドを用いて知識ユニットの編集を行う。なおコマンドMには、より下位レベルのコマンドがある。

```
KNOWLEDGE-UNIT-EDITOR STARTS.  
E-KU>
```

v) ケースデータの編集 (ケースエディタ)

エディタコマンドCを入力すると、以下の様なメッセージが表示されるので、下位コマンドを入力する (表3.12~3.13)。図3.26に実行例を示す。

```
ENTER CASE EDIT COMMAND<I/D/L/P/S/F/B/Q/K>=>
```

vi) 終了

エディタコマンドQUITまたはKILLにより終了する。

表 3.11 知識ユニット編集コマンド

入力形式		機能
I	KUNAME	知識ユニットの新規作成
M	KUNAME C I FF I FH I HH D RULENAME HC HH-RULENAME SI HH-RULENAME SD HH-RULENAME	知識ユニットの適用条件変更 FFルールの追加 FHルールの追加 HHルールの追加 ルールの削除 HHルールの適用条件変更 HHルールのサブルール追加 HHルールのサブルール削除
L	KU	知識ユニットの表示
L	RULE	ルールの表示
D	KUNAME	知識ユニットの削除
Q		エディット内容を保存して終了
K		エディット内容をキャンセルして終了

表 3.12 ケースデータ編集コマンド (その1)

NO.	入力形式	オプション説明	機能
1	I ARG1	ARG1: 生起事象の略称名	データの新規作成
2	D OPT1 OPT2	optionなし	現ポインタのデータを削除
		OPT1: 自然数 OPT2: なし	OPT1で示されるポインタ上のデータを削除
		OPT1: 自然数 OPT2: 自然数	OPT1, OPT2で示される範囲を削除
		OPT1: F OPT2: 自然数	現ポインタからOPT2で示される数のデータを削除
		OPT1: B OPT2: 自然数	現ポインタから逆向きにOPT2で示される数のデータを削除
3	L OPT1 OPT2	optionなし	現ポインタのデータを表示
		OPT1: 自然数 OPT2: なし	OPT1で示されるポインタ上のデータを表示
		OPT1: 自然数 OPT2: 自然数	OPT1, OPT2で示される範囲を表示
		OPT1: F OPT2: 自然数	現ポインタからOPT2で示される数のデータを表示
		OPT1: B OPT2: 自然数	現ポインタから逆向きにOPT2で示される数のデータを表示
	A L L		全データの表示

表3.13 ケースデータ編集コマンド (その2)

NO.	入力形式	オプション説明	機能
4	P		現ポインタの表示
5	S		ポインタを0にセット
6	NUM	NUM:自然数	ポインタをNUMにセット
7	F NUM	NUM:自然数	ポインタを現ポインタ+NUMに セット
8	B NUM	NUM:自然数	ポインタを現ポインタ-NUMに セット
9	Q		エディット内容を保存して終了
10	K		エディット内容をキャンセルして終了

```

ENTER CASE EDIT COMMAND<I/D/L/P/S/F/B/Q/K>=>L

** CASE DATA INFORMATION **

1  NPPA      T      800      NUCLEARE POWER PLANT ACCIDENT

ENTER CASE EDIT COMMAND <I/D/L/P/S/F/B/Q/K>=> L 1 10

** CASE DATA INFORMATION **

1  NPPA      T      800      NUCLEARE POWER PLANT ACCIDENT
2  OTHA      F      0        OTHER ACCIDENT
3  ATRPRE    T      800      REACTOR TRIP
4  ATRPTR    T      800      TURBINE TRIP
5  ATRPGE    T      800      GENERATOR TRIP
6  ACRDSTP   F      0        ROD STOP
7  ACRDBTM   T      800      ROD BOTTOM
8  ACRDWIL   F      0        ROD WITHDRAW LIMIT
9  ALEVPRH   T      801      PRESSURIZER LEVEL HIGH
10 ALEVPRL   F      0        PRESSURIZER LEVEL LOW

```

図 3.26 ケースデータ編集実行例

3.4.5 ロード

知識ベースまたはケースデータのローディングを行なう。

i) メニュー入力

DISKETのトップレベルメニューでLOADを選択する。ファイル名の入力促される。

ii) ファイル名入力

コンパイルされた知識ベース、またはケースデータの保存されているファイル名を記号” (ダブルクォート) で囲んで入力する。

iii) 結果の表示

ローディングが終了すると、メッセージが表示される。図3.27に実行例を示す。

```
# ENTER COMMAND <COMP/ED/LOAD/SAVE/CASE/RUN/LIST/PFD/TEST/TOOL/QUIT>=> LOAD
# ENTER FILE NAME=> "DSNRA(KU1)"

***** LOADING END *****
```

図3.27 ロード実行例

3.4.6 セーブ

コンパイルされた知識ベースまたはケースデータをファイルへ保存する。

i) 知識ベースまたはケースデータの指定

DISKETのトップレベルのメニューでSAVEを選択する。このとき、SAVEに続き、Kと入力すれば、主記憶上にあるコンパイルされた知識ベースが、SAVEに続き、Cと入力すれば、主記憶上にあるケースデータが保存される。

次にファイル名の入力促される。

ii) ファイル名の入力

コンパイルされた知識ベースまたはケースデータを保存するファイル名を入力する。ファイル名は記号”（ダブルクォート）で囲む。

iii) 結果の表示

セーブが終了すると、メッセージが表示される。図3.28に実行例を示す。

```
# ENTER COMMAND <COMP/ED/LOAD/SAVE/CASE/RUN/LIST/PFD/TEST/TOOL/QUIT>=> SAVE
# ENTER OPTIN(K OR C) & FILE-NAME=>K "DSNRA(#KU3)"

# ***** SAVING END *****
```

図3.28 セーブ実行例

3.4.7 推論実行

i) 開始

DISKETのメニュー選択でALL, RUNまたはINFERを選択する。INFERではコンサルテーションは行われぬ。

ii) TIME IDの設定

推論が開始されるとTIME IDの入力メッセージが表示されるので次に示すようにTIME IDを入力する。

```
#ENTER RUN TIME ID => 810
```

入力されたTIME IDが推論実行時の仮想時刻となる。上の例では8時10分である。秒単位の入力が必要な場合は、810.25 (8時10分2.5秒) の様に入力すればよい。

iii) 初期質問の有無

次に推論実行前に質問を行うかどうかきいてくる。

```
# DO YOU USE INITIAL QUESTION MODE? (Y/N)=>
```

ここでYと入力するとFind-TypeがR (ルールから値が決定される生起事象) でない全ての生起事象に関して知識ベースの登録順に質問される。問合せ入力が終了すると推論が開始される。Nと入力すると、ただちにケースファイルを用いて推論が開始される。

iv) 質問入力

推論実行中の問合せには、checklist, multiple choice, yes/No, Numericalの4つのタイプがあり、各タイプに従って入力方法が違う。各タイプ別の問合せ形式と入力例を図3.29に示す。また、表3.14に入力値一覧を示す。

① MULTIPLE CHOICE TYPE

1. TYPE OF PROBLEM:
1 NUCLEAR REACTOR ACCIDENT OCCUR,
2 OTHER PROBLEMS
MULTIPLE CHICE=> 1

② CHECKLIST TYPE

3. SCRAM:
1 SCRAM ACTUATION
2 SCRAM DUE TO HIGH TEMPERATURE OF MAIN COOLANT
3 SCRAM DUE TO LOW FLOW OF MAIN COOLANT
4 SCRAM DUE TO LOW LEVEL OF SG
5 SCRAM DUE TO LOW PRESSURE OF MAIN COOLANT LOOP
6 SCRAM DUE TO SAFETY INJECTION SIGNAL
7 SCRAM DUE TO NEUTRON HIGH FLUX
CHECKLIST=> 1 4

③ YES/NO TYPE

18 SIGNAL ON OF INCONSISTENCY OF PRESSURIZER PRESSURE MEASURING.
YES/NO=> N

④ NUMERICAL TYPE

6 COOLANT FLOW OF A LOOP (%).
NUMERIC=> 51

図 3.29 推論実行時の質問形式

表 3.14 推論実行時の問合せ入力一覧

NO	問合せタイプ	入力値	備 考
1	multiple choice	i	選択可能な唯一の整数を入力する。 選択されたものは真 (True) , その他は偽 (False) とする。
		U	全て未知 (Unknown) とする。
		N	全て偽 (False) とする。
2	checklist	i, j, …… , (UK, m, n, …)	真 (True) の Finding と、"UK" 以降に入力した整数値に対応する Finding は、未知 (unknown) とみなされる。選択されなかったものは偽 (False) となる。() 内は option。
		U	全て未知 (Unknown) とする。
		N	全て偽 (False) とする。
3	numerical	x	Finding の値 x を実数あるいは、整数で指定する。
		U	未知 (Unknown) とする。
4	Yes/No	Y	真 (True) とする。
		N	偽 (False) とする。
		U	未知 (Unknown) とする。

v) 推論結果の呈示

ルールの適用が終了すると推論の結果が自動的に表示される。仮説分類、対応措置、排反生起事象、排反ルールが表示され(図3.30)、続いて図3.31に示すようなメッセージが表示される。

```

** TAXONOMY CERTAINTY (-1-) 1987 NOV. 20,08:10:00

CERTAIN TAXONOMY
FACTOR MNEMONIC EXPLANATION
  0.82 FPCS FAILURE OF PRIMARY COOLANT SYSTEM
  0.32 FPRZ FAILURE OF PRESSURIZER SYSTEM
  0.20 FCVCS FAILURE OF CHEMICAL AND VOLUME CONTROL SYSTEM

** TAXONOMY CERTAINTY (-2-) 1987 NOV. 20,08:10:00

CERTAIN TAXONOMY
FACTOR MNEMONIC EXPLANATION
  0.74 FPCSLP LEAK OF PRIMARY COOLANT LOOP
  0.39 FPRPRC FAILURE OF PRESSURIZER PRESSURE CONTROL SYSTEM
  0.27 FPRHTR FAILURE OF PRESSURIZER HEATER

** TAXONOMY CERTAINTY (-3-) 1987 NOV. 20,08:10:00

CERTAIN TAXONOMY
FACTOR MNEMONIC EXPLANATION
  0.45 RCSLSL SMALL LEAK OF REACTOR COOLANT SYSTEM
  0.43 RCSUTR SG U-TUBE RUPTURE
  0.22 RCRVSO PRESSURIZER RELIEF VALVE STICKS OPEN

I DON'T FIND ANYTHING ABOUT TREATMENT.
NO CONTRADICTORY FINDING EXISTS.
NO CONTRADICTORY RULE IS APPLIED.
FINDINGS ARE ALL CLEAR.

```

図3.30 推論結果

```
# DO YOU SAVE THIS RUN RESULT (Y/N/A)=>
```

図 3.3.1 推論結果保存の問合せ

ここでYまたはAを入力すると、ログファイル（'J2749.1ER.LOG.FILE'）に推論の実行結果が保存される。Yの場合にはファイルの内容は更新（最新の結果のみ保存）されるが、Aの場合には以前の内容に追加が行われる。Nの場合には保存は行われない。

3.4.8 コンサルテーション

i) 開始

推論実行が終了すると以下の様にコンサルテーションが自動的に開始される。

```
# ***** CONSULTATION MODE START *****  
  
# ENTER <QUIT/DR/PC/UF/NR/CR/FIND/FIX/HYPO/SUM/CRF/RULE/RED/RECONF/DETAIL/  
  PRED/RFIND/RFINDC/TRACE/BACK/HELP/STATUS>  
C>
```

メニューを選択することによりコンサルテーションを実行する。表3.15にコンサルテーションメニュー一覧を示す。以下に主なコンサルテーションメニューの使用方法を説明する。（但しTRACEとBACKについては3.4.11節で述べる。）

表 3.15 コンサルテーションメニュー一覧

メニュー	機能
BACK	ルールのバックトレース (知識ベーステストの時のみ使用可)
CR	排反ルールの出力
CRF	排反生起事象の出力
DETAIL	予測コンサルテーション型推論
DR	診断結果の再出力
FIND (生起事象名)	生起事象の値の出力
FIX (生起事象名)	生起事象の値を与える
HELP	HELPメッセージ出力
HYP0 (仮説名)	推論過程の表示
NR	否定ルールの出力
PC	予測コンサルテーション出力コントロール用パラメータの変更
PRED	シミュレーションプログラムの起動
QUIT	予測コンサルテーションの終了
RECONF	生起事象の再確認
RED	ルールの部分修正
RFIND (生起事象名)	指定した生起事象を含むルールの出力
RFINDC (生起事象名)	指定した生起事象を条件部に持つルールの出力
RULE (ルール名)	ルールの内容説明
STATUS	DISKETのシステムステータスの表示
SUM	全生起事象の出力
TRACE	ルールのトレース (知識ベーステストの時のみ使用可)
UF	不明生起事象の出力

ii) DR (診断結果の再出力)

メニューDRを入力すると診断結果が再表示される。

iii) PC (出力コントロールパラメータの変更)

メニューPCを入力すると図 3.32 に示すように、仮説および排反ルールの出力コントロールパラメータの値が出力され、入力が促される。現在は確信度 0.1以上の仮説が表示される。1~4の番号と、パラメータの値を組で入力する。

```

C>PC

ENTER MENU NO. AND CF OF EACH PARAMETER

1 RESET CF OF TAXONOMY CERTAINTY      (0.10)
2 RESET CF OF TREATMENT RECOMENDATION (0.10)
3 RESET CF OF NEGATIVE RULE           (0.10)
4 RESET CF OF RULES CONTRARY TO EACH OTH(0.10)
=> 1 0.4 3 0.3

```

図 3.32 出力コントロール用パラメータの変更

ここで例えば 1 0.4 を入力した場合、図 3.30 の推論結果は図 3.33 の様に変更されて出力される (メニュー DR で出力)。

```

** TAXONOMY CERTAINTY (-1-) 1987 NOV. 20.08:10:00

CERTAIN TAXONOMY
FACTOR  MNEMONIC  EXPLANATION
0.52   FPCS      FAILURE OF PRIMARY COOLANT SYSTEM

** TAXONOMY CERTAINTY (-2-) 1987 NOV. 20.08:10:00

CERTAIN TAXONOMY
FACTOR  MNEMONIC  EXPLANATION
0.74   FPCSLP   LEAK OF PRIMARY COOLANT LOOP

** TAXONOMY CERTAINTY (-3-) 1987 NOV. 20.08:10:00

CERTAIN TAXONOMY
FACTOR  MNEMONIC  EXPLANATION
0.45   RCSSL   SMALL LEAK OF REACTOR COOLANT SYSTEM
0.43   RCSUTR  SG U-TUBE RUPTURE

I DON'T FIND ANYTHING ABOUT TREATMENT.

```

図 3.33 出力コントロール用パラメータ変更後の推論結果出力

iv) UF (不明生起事象の出力)

メニューUFを入力すると、その値が未知であり推論結果に影響を与える生起事象を出力する。(図3.34)

```

C> UF

** IF THE FOLLOWING FINDINGS WERE KNOWN, THE CONCLUSION MIGHT DIFFER.
MNEMONIC  EXPLANATION
AHPCN     ALARM ACTUATION OF HIGH PRESSURE OF CONTAINMENT
DNPCN     SIGNAL OF NORMAL PRESSURE OF CONTAINMENT
AHLVT     ALARM ACTUATION OF HIGH LEVEL OF VOLUME CONTROL TANK
ALLVT     ALARM ACTUATION OF LOW LEVEL OF VOLUME CONTROL TANK
:         :
:         :
:         :

```

図3.34 不明生起事象の出力

v) NR (否定ルールの出力)

メニューNRを入力すると推論で適用された否定ルールが表示される(図3.35)。

```

C> NR

** FOLLOWING RULES ARE APPLIED. (NEGATIVE)

CERTAIN TAXONOMY RULE
FACTOR  MNEMONIC NAME
-0.50   FPCSPM   HS2-1-3

```

図3.35 否定ルールの出力

vi) CR (排反ルールの出力)

メニューCRを入力すると、排反ルール、即ち指定された確信度よりも高い確信度で肯定するルール及び否定するルールが同一の事故仮説に与えられているものがあれば、その事故仮説、確信度、ルール名を表示する(図3.36)。

```

C> CR

** RULES CONTRARY TO EACH OTHER
TAXONOMY  CERTAIN  RULE
MNEMONIC  FACTOR   NAME
POLPP     0.65     FH1-144
          -0.15     FH1-14
          -0.30     FH1-41
RSGBD     0.20     FH1-62
          -0.12     FH1-37
          -0.30     FH1-21
          -0.30     FH1-47
SPBOT     0.42     FH1-124
          -0.35     FH1-46
          -1       FH1-22

```

図 3.36 排反ルールの出力

vii) F I N D (生起事象値の出力)

メニュー F I N D を入力すると、生起事象名の入力促される。以下に実行例 (図 3.37)) を示す。

```

C> FIND
PLEASE ENTER MNEMONIC OF FINDINGS => NPPA
" FIND "
MNEMONIC  VALUE  TIME-ID  EXPLANATION
NPPA      T      800      NUCLEAR POWER PLANT ACCIDENT
          F      0

```

図 3.37 生起事象値の出力

viii) F I X (生起事象値の設定)

メニュー F I X を入力すると、推論実行時に入力設定した時刻における生起事象値を変更できる (図 3.38)。(但しこれは、その時刻以前でかつ最も時刻の近い生起事象値を更新する事。)

```

C> FIX

FIX
PLEASE ENTER MNEMONIC OF FINDINGS => NPPA

PLEASE ENTER VALUE => T
NUCLEAR POWER PLANT ACCIDENT IS UPDATED AS FOLLOWS.
ORIGINAL VALUE =>T
CHANGED VALUE =>T

```

図 3.38 生起事象値の設定

例えば NPPA について (T 800)(F 0) が入力されていたとすると、入力設定時刻が 8 時 10 分の場合、図 3.38 の例では (~~T 800~~)(F 0) と変更される。

ix) H Y P O (推論過程の表示)

メニュー H Y P O を入力すると、指定した仮説における推論過程を表示する。仮説名はメニューと組みで入力する。以下に実行例 (図 3.39) を示す。

```

C> HYPO FPCSLP

HYPO FPCSLP
HISTORY OF <FPCSLP:LEAK OF PRIMARY COOLANT LOOP>
CF    RULE NAME  MNEMONIC&VALUE IN IF-PART OF RULE.
0.50  HS2-1-2   AFLCHAH  F    CHARGING FLOW HIGH
          FLCHAG  22.20CHARGING FLOW (TON HR)
          APREVTL  F    VOLUME CONTROL TANK LOW PRESS.
          VL1460A  F    LET DOWN LINE VALUE LCV 1460A
0.30  HS2-1-5   ALEVPRE  T    PRESSURIZER LEVEL LOW
          APREPRL  T    PRESSURIZER PRESSURE LOW
0.25  CF IMPLIED BY PREDECESSOR POSITIVE
0.00  CF IMPLIED BY PREDECESSOR NEGATIVE

```

図 3.39 推論過程の表示

x) SUM (全生起事象の出力)

メニューSUMを入力すると、全生起事象を表示する。以下に実行例(図3.40)を示す。

```

C> SUM

** FINDINGS SUMMARY

MNEMONIC EXPLANATION

TYPE OF PROBLEM
  NPPA      NUCLEAR POWER PLANT ACCIDENT      IS TRUE.      800
                                                    IS FALSE.     0
  OTHA      OTHER ACCIDENT                    IS FALSE.     0

ANNUNCIATORS OF TRIP
  ATRPRE    REACTOR TRIP                      IS FALSE.     0
  ATRPTR    TURBINE TRIP                      IS FALSE.     0
  ATRPGE    GENERATOR TRIP                    IS FALSE.     0
  :         :                                 :             :
  :         :                                 :             :

ANNUNCIATORS OF PRESSURIZER
  ALEVPRH   PRESSURIZER LEVEL HIGH           IS FALSE.     0
  ALEVPRL   PRESSURIZER LEVEL LOW            IS TRUE.      801.0
                                                    IS FALSE.     0
  APREPRH   PRESSURIZER PRESSURE HIGH        IS FALSE.     0
  APREPRL   PRESSURIZER PRESSURE LOW         IS TRUE.      802.20
                                                    IS FALSE.     0
  :         :                                 :             :
  :         :                                 :             :
  :         :                                 :             :

```

図3.40 全生起事象の出力

xi) CRF (排反生起事象の出力)

メニューCRFを入力すると、推論過程で排反生起事象、即ちFFルールに反する生起事象があればその生起事象とルールを表示する(図3.41)。

```

C> CRF

** FOLLOWING FINDINGS ARE CONTRADICTIONARY
1 APREPRL PRESSURIZER PRESSURE LOW
  T : FF1-7      F(PREPZR,#.0:154) -> F(APREPRL,T)
  F : TABLE
2 ATEMLDH LETDOWN FLOW HIGH TEMP.
  T : FF1-16     F(TEMLD,#.193:*) -> F(ATEMLDH,T)
  F : TABLE (=) インプットデータである事を示す
          :
          :
          :

```

図 3.4 1 排反生起事象の出力

xii) RULE (ルールの内容表示)

メニューRULEを入力すると、ルール名の入力促される。以下に実行例(図3.42)を示す。

```

C> RULE
  ENTER RULE NAME => HS1-1-1

HH RULE
  HS1-1-1 : (1:H(FPCS,0.10:1.0),H(FPRZ,0.10:1.0)) -> H(IHTAX1.0.40)

```

図 3.4 2 ルールの内容説明

xiii) RED (ルールの部分修正)

メニューREDを入力すると、RULE名と変更の指定の入力が促される。図3.43に示す例ではルールHS1-1-1のうち、最初のFが、Tに変更された。変更の指定は、記号”(ダブルクォート)で囲う。

```

C> RED
ENTER <RULENAME & STR1 & STR2> -> HS1-1-1 ".F" ".T"
    HS1-1-1      : (1:F(YREVFL1,F),F(YREVFL2,F),F(YREVFL3,F))&(1:F(ALEVPRH,F)
                  ,F(APREPRH,F)) -> H(RCSPL1,0,80)
    HS1-1-1      : (1:F(YREVFL1,F),F(YREVFL2,F),F(YREVFL3,F))&(1:F(ALEVPRH,F)
                  ,F(APREPRH,F)) -> H(RCSPL1,0,80)

```

図 3.43 ルールの部分修正

xiv) RECONF (生起事象の再確認)

メニュー RECONF を入力すると次の様なメッセージが表示される (図 3.44)。

```

C> RECONF
SELECT FROM FOLLOWING CONSULTATIONS
1 RECONFIRM UNKNOWN FINDINGS
2 RECONFIRM SPECIFIC FINDING
3 RECONFIRM ALL FINDINGS
ENTER MENU-NO./CAN
>

```

図 3.44 RECONF の出力メッセージ

1 を入力すると値不明の生起事象について、2 を入力すると指定した生起事象について (図 3.45)、3 を入力する図 3.46 に示すようにと全ての生起事象について再確認が行われる。

```

> 2
APPOINT THE NAME OF FINDINGS
ENTER FINDING-NAME(MNEMONIC)/CAN
> OTHA
RECONFIRMATION INFORMATION
    OTHA      OTHER ACCIDENT                      IS FALSE. 0
** RECONFIRMATION INFORMATION
1 OTHA      OTHER ACCIDENT
    T :
    F : TABLE

```

図 3.45 指定生起事象の再確認


```

> 3
RECONFIRMATION INFORMATION
  NPPA      NUCLEAR POWER PLANT ACCIDENT      IS TRUE.   810
                                                    IS FALSE.   0
  OTHA      OTHER ACCIDENT                    IS FALSE.   0
  ATRPRE    REACTOR TRIP                      IS FALSE.   0
  :         :                                  :         :
  :         :                                  :         :
  ALEVPRH   PRESSURIZER LEVEL HIGH            IS FALSE.   0
  ALEVPRL   PRESSURIZER LEVEL LOW            IS TRUE.   801.0
                                                    IS FALSE.   0
  APREPRH   PRESSURIZER PRESSURE HIGH        IS FALSE.   0
  APREPRL   PRESSURIZER PRESSURE LOW        IS TRUE.   802.20
                                                    IS FALSE.   0
  :         :                                  :         :
  :         :                                  :         :
  :         :                                  :         :
  
```

図 3. 4 6 全生起事象の再確認

xv) D E T A I L (詳細事故診断)

メニュー D E T A I L を入力すると次のメッセージが表示される (図 3. 4 7)。
1 ~ 4 を選択する。

```

C> D E T A I L
SELECT FROM FOLLOWING CONSULTATIONS
1 RESEARCH IN DETAIL
2 RESEARCH SPECIFIC TAXONOMY IN DETAIL
3 RESEARCH ABOUT TAXONOMY AS RESULT
4 RESEARCH ABOUT ALL TAXONOMY
ENTER MENU-NO./CAN
>
  
```

図 3. 4 7 D E T A I L の出力メッセージ

以下、それぞれのメニューについて実行結果を示す。1は包括推論の結果最大の確信度を得た仮説について（図3.48）、2は指定した仮説について（図3.49）、3は、推論された全仮説のうちから確信度の高い順に（図3.50）推論過程を表示し、4は診断結果の表示を行う（図3.51）。

```

ENTER MENU-NO./CAN
> 1

** DETAIL INFORMATION
HISTORY OF <PSPV01:PRESSURIZER SPRAY CONTROL ONE FAILS OPEN> CF = 0.91
CF  RULE NAME  MNEMONIC&VALUE IN IF-PART OF RULE.
0.90 HS3-3-1   SPRZPL   F   PRESSURIZER LOW PRESSURE
      APREPRL  T   PRESSURIZER PRESSURE LOW
0.10 CF IMPLIED BY PREDECESSOR POSITIVE
0.00 CF IMPLIED BY PREDECESSOR NEGATIVE
    
```

図3.48 詳細事故診断の実行結果（その1）

```

ENTER MENU-NO./CAN
> 2
APPOINT THE NAME OF TAXONOMY
ENTER TAXONOMY-NO./CAN
> RCSLSL

** DETAIL INFORMATION
HISTORY OF <RCSLSL:SMALL LEAK OF REACTOR COOLANT SYSTEM> CF = 0.59
CF  RULE NAME  MNEMONIC&VALUE IN IF-PART OF RULE.
0.30 HS3-2-4   AFLCHAN  F   CHARGING FLOW HIGH
      FLCHAG   22.20CHARGING FLOW (TON HR)
0.24 CF IMPLIED BY PREDECESSOR POSITIVE
0.00 CF IMPLIED BY PREDECESSOR NEGATIVE
    
```

図3.49 詳細事故診断の実行結果（その2）

```

ENTER MENU-NO./CAN
> 3

** DETAIL INFORMATION
HISTORY OF <PSPV01:PRESSURIZER SPRAY CONTROL ONE FAILS OPEN> CF = 0.91
CF    RULE NAME  MNEMONIC&VALUE IN IF-PART OF RULE.
0.90  HS3-3-1   SPRZPL   F   PRESSURIZER LOW PRESSURE
        APREPRL  T   PRESSURIZER PRESSURE LOW
0.10  CF IMPLIED BY PREDECESSOR POSITIVE
0.00  CF IMPLIED BY PREDECESSOR NEGATIVE

** DETAIL INFORMATION
HISTORY OF <FPCS:FAILURE OF PRIMARY COOLANT SYSTEM> CF = 0.82
CF    RULE NAME  MNEMONIC&VALUE IN IF-PART OF RULE.
0.60  FH1-8     ARADMAL  T   RADIATION MONITOR ALARM
0.35  FH1-7     AFLCHAH  F   CHARGING FLOW HIGH
        FLCHAG   22.20CHARGING FLOW (TON HR)
0.30  FH1-10    ALEVPRL  T   PRESSURIZER LEVEL LOW
        APREPRL  T   PRESSURIZER PRESSURE LOW
0.00  CF IMPLIED BY PREDECESSOR POSITIVE
0.00  CF IMPLIED BY PREDECESSOR NEGATIVE

** DETAIL INFORMATION
HISTORY OF <FPCSLP:LEAK OF PRIMARY COOLANT LOOP> CF = 0.80
:      :      :
:      :      :
:      :      :

```

図 3.50 詳細事故診断の実行結果 (その3)

ENTER MENU-NO./CAN

> 4

** TAXONOMY CERTAINTY (-1-) 1987 NOV. 20,08:10:00

CERTAIN TAXONOMY

FACTOR MNEMONIC EXPLANATION

0.82 FPCS FAILURE OF PRIMARY COOLANT SYSTEM

0.32 FPRZ FAILURE OF PRESSURIZER SYSTEM

0.20 FCVCS FAILURE OF CHEMICAL AND VOLUME CONTROL SYSTEM

** TAXONOMY CERTAINTY (-2-) 1987 NOV. 20,08:10:00

CERTAIN TAXONOMY

FACTOR MNEMONIC EXPLANATION

0.74 FPCSLP LEAK OF PRIMARY COOLANT LOOP

0.39 FPRPRC FAILURE OF PRESSURIZER PRESSURE CONTROL SYSTEM

0.27 FPRHTR FAILURE OF PRESSURIZER HEATER

** TAXONOMY CERTAINTY (-3-) 1987 NOV. 20,08:10:00

CERTAIN TAXONOMY

FACTOR MNEMONIC EXPLANATION

0.45 RCSSL SMALL LEAK OF REACTOR COOLANT SYSTEM

0.43 RCSUTR SG U-TUBE RUPTURE

0.22 RCRVSO PRESSURIZER RELIEF VALVE STICKS OPEN

I DON'T FIND ANYTHING ABOUT TREATMENT.

NO CONTRADICTIONARY FINDING EXISTS.

NO CONTRADICTIONARY RULE IS APPLIED.

FINDINGS ARE ALL CLEAR.

図 3.5 1 詳細事故診断の実行結果 (その 4)

xvi) P R E D (シミュレーションプログラムの起動)

メニューP R E Dを入力するとシミュレーションプログラム (現在はPWRDYN) の起動を行う事ができる (図3. 5 2)。

```
C> PRED
# DO YOU EXECUTE <PWRDYN> <Y/N>=> Y
KEQ56250I JOB JCLG(JOB04268) SUBMITTED
SIMULATION FINISHED
```

図3. 5 2 シミュレーションプログラムの起動

xvii) R F I N D (指定生起事象を含むルールの出力)

メニューR F I N Dを入力すると、指定した生起事象を含むルールが表示される (図3. 5 3)。

```
C> RFIND
ENTER MNEMONIC NAMES => NPPA
# NUCLEAR POWER PLANT ACCIDENT(NPPA) IS USED AS FOLLOWS.
# 'NPPA' IS NOT USED IF PART OF ANY RULE.
# 'NPPA' IS NOT USED THEN PART OF ANY RULE.
# NPPA IS USED BY THE CONDITION OF FOLLOWING HH RULES.
1  HH4-1      F(NPPA,T)
2  HH1-1      F(NPPA,T)
```

図3. 5 3 指定生起事象を含むルールの出力

xviii) R F I N D C (指定生起事象を条件部を含むルールの出力)

メニューR F I N D Cを入力すると、指定した生起事象を条件部を含むルールが表示される (図3. 5 4)。生起事象名はメニューと組みで入力する。

```
C> RFINDC ATEPRTH
"IF PARTS OF NEXT RULES USE FOLLOWING FINDINGS"
ATEPRTH      :  RULE => (FH1-12 FH1-18 HS2-2-3)
```

図3. 5 4 指定生起事象を条件部を含むルールの出力

xix) STATUS (システムステータスの表示)

メニューSTATUSを入力すると、推論実行CPU TIME, 適用されたルール数, 成立したルール数およびガベジコレクション回数が表示される (図3.55)。

```

C> STATUS

*** IERIAS SYSTEM STATUS ***
SESSION CPU TIME      = 10226
INFERENCE CPU TIME    = 887
NUMBER OF EVAL RULES  = 190
NUMBER OF ACTION RULES = 21
NUMBER OF GC          = 1
  
```

図3.55 システムステータスの表示

3.4.9 計算コードの制御

DISKETでは、事故診断の結果に基づいたシミュレーションコード (現在はPWR DYNを使用) を実行させる事ができる。入力データの値, 時刻, 入力形式はルールにより自動的に作成させる。図3.56に計算コードを利用する場合に必要な知識ベースの表現方法の例を示す。

```

***ACTION DEFINITION
**ACTION PREDICT
*LANGUAGE L
*FORMAT PWRDYN-DATA
(D*SIMULATION)
***FORMAT DEFINITION
**OUTPUT FORMAT
*NAME
PWRDYN-DATA
*TYPE
FILE
*FILENAME
'J2749.PWRDYN.DATA'
*ORIGINAL
'J2749.PWRDYN.ORI.DATA'
*FORMAT
F1 (8F10.1)
F2 (15I5)
*OUTPUT
(COPY UNTIL (MATCH "CALCULATION CONDITION"))
(OUTPUT "*" Q")
(OUTPUT "F1 PD-Q)
(OUTPUT "*" KTCV KACS KSCRAM KWF KPTRI KRFP KRFS KSDS KSAT KSPI KHOT")
(OUTPUT F1 0 0 2 3 PD-KPTRIP PD-KRFPR PD-KRFSG PD-KSDSG 1 0 0)
:
:
:

```

図 3.56 計算コードを利用するときの知識ベースの表現方法

```

      :
      :
      :
(IF (EQ PD-KCHANGE1 1)
  (THEN (OUTPUT "* KSET(1) ATRPTR F -> T")
        (OUTPUT F2 4)
        (OUTPUT F1 0.0 TIME-ATRPTR-T TIME-ATRPTR-T+0.2
                  TIME-ATRPTR-T+50)
        (OUTPUT F1 0.0 0.0 -0.1 -0.1)))
      :
      :
      :
(OUTPUT "* PLOTTER DATA -----")
(OUTPUT "* DPLOT  DPLOT1  TIMPLT  DPRPLT  NPLOT  KGH")
(OUTPUT " 0.2    0.2    80.0    0.0    0    0")
(OUTPUT "**")
(OUTPUT "*")

```

図 3.56 (続き)

3.4.10 知識ベースの検索

DISKETのトップレベルのメニューでLISTを選択すると、表3.16に示されるLISTコマンドの入力が促される。それぞれのコマンドにより、知識ベースの内容を表示できる。なお図3.57～3.64に各コマンドの実行例を示す。

表3.16 LISTコマンド

コマンド	対 象
FIND	生起事象
HYP0	仮説
IH	中間仮説
KB	知識ベース
KU	知識ユニット
RULE	ルール
TAX	仮説分類 (事故仮説)
TRT	対応措置 (対応仮説)

```
#ENTER LIST COMMAND<HYPO/TAX/IH/TRT/FIND/KB/KU/ALL/RULE>=> HYPO
```

DATA DEFINITION LIST

** HYPOTHESIS

* TAXONOMY

1 NRA	:	NUCLEAR POWER PLANT ACCIDENT
2 FPCS (0.18)	:	FAILURE OF PRIMARY COOLANT SYSTEM
3 FPCSPM (0.60)	:	FAILURE OF PRIMARY COOLANT PUMP
4 RCSPL1 (0.30)	:	LOSS OF REACTOR COOLANT ONE PUMP
4 RCSPL2 (0.20)	:	LOSS OF REACTOR COOLANT TWO PUMPS
4 RCSPLA (0.20)	:	LOSS OF REACTOR COOLANT ALL PUMPS
:	:	:
:	:	:
:	:	:

* INTERMEDIATE HYPOTHESIS

1 IHTAX1	:	APPLIED CONDITION OF KNOWLEDGE UNIT 2
1 IHTAX2	:	APPLIED CONDITION OF KNOWLEDGE UNIT 3
1 FRADM	:	FAILURE OF RADIATION MONITER

* TREATMENT RECOMMENDATION

1 TREAT01	:	TAKE ACTION OF TREATMENT NO.1
1 TREAT02	:	TAKE ACTION OF TREATMENT NO.2
:	:	:
:	:	:
:	:	:

図 3.57 仮説の検索の実行例

```

#ENTER LIST COMMAND<HYPO/TAX/IH/TRT/FIND/KB/KU/ALL/RULE>=> TAX

DATA DEFINITION LIST

* TAXONOMY

1 NRA                : NUCLEAR POWER PLANT ACCIDENT
2 FPCS (0.18)        : FAILURE OF PRIMARY COOLANT SYSTEM
3 FPCSPM (0.60)      : FAILURE OF PRIMARY COOLANT PUMP
4 RCSPL1 (0.30)      : LOSS OF REACTOR COOLANT ONE PUMP
4 RCSPL2 (0.20)      : LOSS OF REACTOR COOLANT TWO PUMPS
4 RCSPLA (0.20)      : LOSS OF REACTOR COOLANT ALL PUMPS
:                    :
:                    :
:                    :

```

図 3.58 仮説分類（事故仮説）の検索の実行例

```

#ENTER LIST COMMAND<HYPO/TAX/IH/TRT/FIND/KB/KU/ALL/RULE>=> IH

DATA DEFINITION LIST

* INTERMEDIATE HYPOTHESIS

1 IHTAX1             : APPLIED CONDITION OF KNOWLEDGE UNIT 2
1 IHTAX2             : APPLIED CONDITION OF KNOWLEDGE UNIT 3
1 FRADM              : FAILURE OF RADIATION MONITER

```

図 3.59 中間仮説の検索の実行例

```
#ENTER LIST COMMAND<HYPO/TAX/IH/TRT/FIND/KB/KU/ALL/RULE>=> TRT

DATA DEFINITION LIST

* TREATMENT RECOMMENDATION
! TREAT01           : TAKE ACTION OF TREATMENT NO.1
! TREAT02           : TAKE ACTION OF TREATMENT NO.2
:                   :
:                   :
:                   :
```

図 3.6 0 対応措置（対応仮説）の検索の実行例

```
#ENTER LIST COMMAND<HYPO/TAX/IH/TRT/FIND/KB/KU/ALL/RULE>=> FIND

FINDINGS

1. ANNUNCIATORS OF TRIP
: CHECKLIST           FTYPE=E  NT=0
(1)ATRPRE           : REACTOR TRIP
(2)ATRPTR           : TURBINE TRIP
(3)ATRPGE           : GENERATOR TRIP

2. ANNUNCIATORS OF CONTROL ROD
: CHECKLIST           FTYPE=E  NT=2
(1)ACRDSTP          : ROD STOP
(2)ACRDBTM          : ROD BOTTOM
(3)ACRDWIL          : ROD WITHDRAW LIMIT
:                   :
:                   :
:                   :
```

図 3.6 1 生起事象の検索の実行例

```

#ENTER LIST COMMAND<HYPO/TAX/IH/TRT/FIND/KB/KU/ALL/RULE>=> KB
          KNOWLEDGE BASE LIST
*** KNOWLEDGE UNIT (1) (KU1)
**  UNIT CONDITION          : (1)
**  MESSAGE OPTION         : NIL
**  FF RULES
    FF1-1      : F(PRCNT,#,0.62:*) -> F(AINISG,T)
      :
      :
**  FH RULES
    FH1-1      : F(FLCHAG,#,15:*)&F(ARADMAL,T) -> H(FRADM,-0.10)
      :
      :
**  HH RULES
*  HH RULE 1 (HH1-1)
    IF : (F NPPA T)
    THEN : HS1-1-1      (1:H(FPCS,0.10:1.0),H(FPRZ,0.10:1.0)) -> .....
      :
      :
*** KNOWLEDGE UNIT (2) (KU2)
**  UNIT CONDITION          : (((H IHTAX1 +0.4000000--+00 +0.1000000--+01}))
**  MESSAGE OPTION         : NIL
**  FF RULES
**  FH RULES
**  HH RULES
*  HH RULE 1 (HH2-1)
      :
      :

```

図 3.6 2 知識ベース表示の実行例

```

#ENTER LIST COMMAND<HYPO/TAX/IH/TRT/FIND/KB/KU/ALL/RULE>=> KU

KU-NAME? KU1

                KNOWLEDGE BASE LIST

*** KNOWLEDGE UNIT (I) (KU1)
** UNIT CONDITION      : (1)
** MESSAGE OPTION     : NIL
** FF RULES
    FF1-1      : F(PRCNT, #, 0.62:*) -> F(AINISG, T)
    FF1-2      : F(LCRODD, #, 0:10) -> F(ACRDBTM, T)
    FF1-3      : F(LCRODD, #, 220:*) -> F(ACRDWIL, T)
    :          :
    :          :
** FH RULES
    FH1-1      : F(FLCHAG, #, 15:*)&F(ARADMAL, T) -> H(FRADM, -0.10)
    FH1-2      : F(SPRZPL, T) -> H(FPCS, 0.05)
    FH1-3      : F(!:F(SRCSL1, T), F(SRCSL2, T)) -> H(FPCS, 0.50)
    :          :
    :          :
** HH RULES
* HH RULE 1 (HH1-1)
  IF   : (F NPPA T)
  THEN : HS1-1-1      (!:H(FPCS, 0.10:1.0), H(FPRZ, 0.10:1.0)) -> .....
    :   :
    :   :
    :   :

```

図 3.63 知識ユニットの検索の実行例

```
#ENTER LIST COMMAND<HYPO/TAX/IH/TRT/FIND/KB/KU/ALL/RULE>=> RULE

ENTER RULE NAME => FF1-1

FF RULE
    FF1-1      : F(PRCNT, #.0, 62:*) -> F(AINISG, T)
```

図 3.6 4 ルールの検索の実行例

3.4.1.1 知識ベースのテスト

i) ブレークポイントの指定

DISKETのトップレベルメニューでTESTを選択すると次のメッセージが表示される。

```
ENTER BREAK COUNT ('N/ALL)-->
```

ここで、ALLと入力すると通常の包括型推論を実行した後、コンサルテーションに入り、メニューTRACEとBACKが使用できる。続けて一度トップレベルメニューに戻った後、再びTESTメニューの選択、ブレークポイント指定でn POSEと入力すると、n回ルールが適用されるたびに推論が中断され、メニューTRACEとBACKが使用できる。

ii) テストのメニュー

コンサルテーションメニューに含まれているTRACEとBACKは知識ベースのテストのときのみ使用できる。

① TRACE (ルールのトレース)

ルールを適用された順に表示する(図3.65)。

② BACK (ルールのバックトレース)

ルールを最後に適用されたものから逆の順に表示する(図3.66)。

```

C> TRACE
** IERIAS RULEAPPLIED HISTORY **
FF1-1  -1  (NONAPPLIED)
          FF1-1      : F(PRCNT,0.62:*) -> F(AINISG,T)
FF1-2  -1  (APPLIED)
          FF1-2      : F(LCRODD,0.10) -> F(ACROBTM,T)
FF1-3  -1  (NONAPPLIED)
          FF1-3      : F(LCRODD,220:*) -> F(ACROWIL,T)
FF1-4  -1  (APPLIED)
          FF1-4      : F(LEVPRT,0.05:*) -> F(ALEVPRH,T)
FF1-5  -1  (NONAPPLIED)
          FF1-5      : F(LEVPRT,-1:-0.06) -> F(ALEVPRL,T)
FF1-6  -1  (NONAPPLIED)
          FF1-6      : F(PREPRZ,163:*) -> F(APREPRH,T)
FF1-7   1  (APPLIED)
          FF1-7      : F(PREPRZ,0:154) -> F(APREPRL,T)
FF1-8   1  (APPLIED)
          FF1-8      : F(TEMPRL,77:*) -> F(ATEPRLH,T)
FF1-9   1  (APPLIED)
          FF1-9      : F(TEMPRT,52:*) -> F(ATEPRTH,T)
FF1-10  1  (APPLIED)
          FF1-10     : F(PREPRT,0.20:*) -> F(APRERTH,,T)

```

図 3.65 ルールのトレース

C> BACK

** IERIAS RULE APPLIED HISTORY , BACKWARD CHAINING **

FH1-25 1 (APPLIED)

FH1-25 : F(VL1460A,F) -> H(FCVCS,0.20)

FH1-24 -1 (NONAPPLIED)

FH1-24 : (1:F(APREVTL,T),F(APREVTH,T)) -> H(FCVCS,0.30)

FH1-23 1 (APPLIED)

FH1-23 : (1:F(AFLCHAH,T),F(AFLCHAL,T)) -> H(FCVCS,0.30)

FH1-22 1 (APPLIED)

FH1-22 : F(ACRDBTM,T) -> H(FCRD,0.05)

FH1-21 -1 (NONAPPLIED)

FH1-21 : F(ACRDSTP,T) -> H(FCRD,0.25)

FH1-20 1 (APPLIED)

FH1-20 : F(ATEAVRF,T) -> H(FCRD,0.05)

FH1-19 -1 (NONAPPLIED)

FH1-19 : F(ACRDWIL,T) -> H(FCRD,0.30)

FH1-18 -1 (NONAPPLIED)

FH1-18 : F(SPRFXH,T) -> H(FCRD,0.30)

FH1-17 1 (APPLIED)

FH1-17 : (1:F(ATEPRLH,T),F(ATEPRLH,T),F(APRERTH,T)) -> H(FPRZ,0.15)

FH1-16 1 (APPLIED)

図 3.66 ルールのバックトレース

3. 5 操作上の注意

i) ファイル名の入力

本システムで、端末からファイル名を入力するときは、記号”（ダブルクォート）で囲まなければならない。

```
“ 'J2749.RCSUTR.DATA' ”
```

ii) 知識ベース構築支援機能の回復

知識ベース構築支援機能の操作中、誤操作等により次の様なメッセージが表示されることがある（UTILISPのトップレベルに戻る）。

```
DISKET ERROR OCCURED  
>
```

この時は

```
(DS*ED-TOP)
```

と入力すれば、知識ベース構築支援初期画面が表示される。但し、入力は始めからやり直す必要がある。

3.6 エラーメッセージ一覧

DISKETに用意されているエラーメッセージを以下にしめす。

DISKET エラーメッセージ (その1)

注 メッセージの先頭に、エラーコード (S1001など) がつくことがある。

# this file doesn't exist→ファイル名	コンパイルする (ファイル名の) なし。
# illegal header notation	知識ベース定義のヘッダーが***DATAではない。
# hypothesis not found	仮説がない。
# illegal input→入力	入力エラー (知識ベース中の、* 1つのもの)
# reserved word used→略称	略称に予約語が使うわれている。
# this mnemonic already exist→略称	略称を重複登録しようとした。
# not compiled next image→	仮説定義中の次の表現がコンパイルできない。
# table list unmatched	コンパイル使用テーブルの整合性がとれていない。
# weight data not found 0.25 generated→	階層的仮説に重みづけがされていないので、0.25が割りふられた。
# findings type error !!! next line is not compiled --> 事象入力	事象として入力されたもののコンパイルできない。
# no header	ヘッダー (***...) がない。
# illegal find-type →	値設定タイプに i, l, r 以外のものがある。
# this header has no finding→	宣言文のあとに、事象が定義されていない。
# not compiled next line→	事象定義における次の表現がコンパイルできない。
# no findings -	数値型事象の宣言文があるが、事象名が定義されていない。
# illegal defined range→	数値型事象の値域指定にエラーあり。

DISKET エラーメッセージ (その2)

# illegal knowledge base header!!	知識ベース宣言文がない。
# illegal knowledge unit header	知識ユニット宣言文がない。
# this knowledge unit already exists	この知識ユニットは既に定義されている。
# mnemonic or value or fh is illegal→	ルールのシンタックスエラーあり。
# mnemonic or value is not found→	ルールのシンタックスエラーあり。
# this finding's mnemonic or value is illegal→	ルール定義中の事象と値の 組合せのエラー
# this hypothesis' mnemonic or value is illegal→	ルール定義中の仮説と値の組合せ のエラー
# action format error→	実行処理定義のエラー
# message format error→	メッセージ表示ルール定義のエラー
# then part is not found	ルールの実行部がない。
# then part format is illegal→	ルール実行部のフォーマットエラー
# then part mne syntax error→	"
# format missing	ルール実行部のデータ指定のエラー
# * if statement not found	HHルールの* if 宣言文がない。
# * then statement not found	HHルールの* then 宣言文がない。
# hh rule condition not found	HHルールの適用条件がない。
# knowledge unit not found	知識ユニットがない。
# illegal time id , please retry	Time Id 入力のエラー
# this fname is not finding→	事象のエラー
# case file name error→	ケースファイル名のエラー
# illegal rule format	ルールフォーマットのエラー
# input illegal , please enter again!!	入力エラー
# illegal file name , please retry!	ファイル名入力エラー
# illegal option ! please reenter	選択値エラー
# file name illegal	ファイル名入力エラー
# illegal input range	数値型事象への入力の値域エラー
# illegal action list	ルールタイプのエラー
# illegal input found ,please retry!!!	事象値の入力エラー
# mne duplicate , this input is ignored	ケースデータ作成時の入力エラー
# illegal input , this input is ignored	"
# length error , this input is ignored	"

DISKET エラーメッセージ (その3)

# format error	ケースデータのフォーマットエラー
# value error	ケースデータタイプのエラー
# range error	数値型ケースの値域エラー
# if part is not found	ルールの条件部がない。
# if part first element is illegal	ルールの条件部のエラー
# mnemonic name syntax is illegal→	ルールの条件部の略称と値の結合のエラー
# after rule , format illegal	after ルールのフォーマットエラー
# this knowledge unit has not ff-rule→	知識ユニットに FF ルールがない。
# rule format error →	ルールのフォーマットエラー
# this knowledge unit has not fh-rules→	知識ユニットに FH ルールがない。
# illegal unit condition , generate 1	適用条件がおかしい。1が採用された。
# unit condition not found , generate 1	適用条件がない。1が採用された。
# this finding is not found	入力された事象が知識ベース上に存在しない。
# illegal value	事象の入力値エラー
data edit command not entered	ケースエディターコマンドが入力されていない。
# illegal input command	ケースエディターコマンドのエラー
# illegal 2nd option	ケースエディターコマンド入力エラー
# this mnemonic is not hypothesis	略称が仮説ではない。
# this mnemonic has never defined	略称が未定義である。
# options are not entered , please retry!	選択肢が入力されていない。
# illegal options , please retry!	選択肢入力エラー
# illegal command input	LIST コマンドのエラー
# edit command option error	ケースエディターコマンドの 選択肢入力エラー
# illegal input found	入力エラー
# hh rule condition not found	HH ルールの適用条件がない。
# mnemonic name illegal !!!	略称のエラー
# illegal argument , please retry!!!	コマンドの引数のエラー
# illegal range of data	値域のエラー
# option error	選択肢のエラー
case data does not exist!!!	ケースデータがない。
# input rule format illegal!!!	ルールのフォーマットエラー

DISKET エラーメッセージ (その4)

# illegal knowledge unit no!!!	知識ユニット番号のエラー
# illegal input . please retry	一価選択型質問に対して、複数の入力を行った。
# rule input is canceled!!!	ルール修正がキャンセルされた。
*** input error **	ルールタイプの入力エラー
# illegal input format!!!	ルール修正の入力エラー
# rule modification is cancelled!!!	ルール修正がキャンセルされた。
# → is not found!!!	ルール定義中に→がない。
** sub-rule does not exist	サブルールがない。
** sub-rule name illegal	サブルール名のエラー
**rule does not exist	ルールがない。
# illegal knowledge unit header!!	知識ユニット宣言文のエラー
# action name not found!!!	実行処理宣言文がない。
% this line is not compiled→	コンパイルエラー
# :this action has not method→	実行処理定義の実行処理部がない。
# next action function is not define d →	実行処理定義中の LISP 関数が定義されていない。
略称 is not mnemonic	指定された仮説が存在しない。
taxonomy not found	知識ベース上に仮説が定義されていない。
** 略称 is not a hypothesis or does not have conclusion	指定された仮説が存在しない。
method type is illegal!!!	実行処理定義中の使用言語指定のエラー
illegal value found!!!	事象値のエラー
# 略称 is not decided	指定された事象が値を持っていない。
this finding's value is not decided→	略称
this input is not finding→	略称 指定された略称は事象ではない。
# your input is illegal , please retry	トップレベルコマンド入力のエラー
disket system error ! & recovered!	エラーが発生したが回復された。

参考文献

- 1) Weiss, S.M. and Kulikowski, C.A.: "EXPERT:A System for development Consultation Models", CBM-TR-97 (1979)
- 2) 岡田二郎, 横林正雄, 溝口文雄 「知識工学的手法による原子炉事故診断システムの開発」 日本原子力学会誌 Vol.25, No.6 P469~479 (1983)
- 3) Yokobayashi, M., Yoshida, K., Kohsaka, A. and Yamamoto, M. : Development of Reactor Accident Diagnostic System DISKET using Knowledge Engineering Technique, J.Nucl.Sci.Technol., 23[4], (1986)
- 4) Yoshida, K., Yokobayashi, M., Aoyagi, T., Shinohara, Y. and Kohsaka, A. : Development and Verification of an Accident Diagnostic System for Nuclear Power Plant by using a Simulator. ANS Topical Meeting on Computer Applications for Nuclear Power Plant Operation and Control, Pasco, USA September (1985)
- 5) 横林正雄, 吉田一雄, 山本 稔, 鴻坂厚夫 「知識工学を用いた原子炉事故診断システムの推論機構: I E R I A S」 JAERI-M 84-205 (1984)
- 6) 横林正雄, 山本 稔, 吉田一雄 「知識工学を用いた原子炉事故診断時の予測コンサルテーション機能の開発」 日本原子力学会分科会 A-47 (1984)

付録A UTILISPの概要

A. 1 機能概要

UTILISPは、対話型言語プロセッサであり、記号処理 リスト処理だけでなく、ソフトウェアツールの作成など、単純大規模数値計算以外のあらゆる用途に使用可能である。

言語仕様は、MACLISPの系統のLisp-Machine-Lispに類似しており、以下のような特徴を持っている。

- ・比較的低いレベルの機能がユーザに解放されている。
- ・文字列処理が強力である。
- ・マクロ機能がある。
- ・高速である。
- ・各種のユティリティがある。

ユティリティには、次のものがある。

- ・コンパイラ
- ・構造指向エディタ
- ・トレーサ
- ・プリティプリンタ
- ・プリティリーダー

A. 2 ソフトウェア組合せ条件

本ソフトウェアは、下記の環境で動作します。なお、関連ソフトウェアの提供スケジュールにご注意下さい。

(1) 適用条件

本ソフトウェアは以下のシステムのもとで使用することができる。

OSIV/F4 システム制御プログラム：E40以降 (TSS)

OSIV/F4 MSP システム制御プログラム：E10以降 (TSS)

(2) 提供媒体形式と内容

本ソフトウェアで、提供される磁気テープの内容とDASD上での大きさは下記のとおりです。なお、DASD上での大きさはF479で算出しています。

表A. 1 磁気テープの内容とDASD上での大きさ

種別	記録密度	ラベル形式	ボリューム 通し番号		データセット 数	サイズ	巻数
DLIB	1600RPI (9トラック)	1. SL ② NL	UTLISP		8 ÷ α	1. FULL 2. HALF ③ QUARTER	1
格納 順 番	データセット名	レ コ ド 形 式	レ コ ド 長	ブ ロ ック 長	内 容	DASD上での 大きさと形式	
						大きさ TRK	データセ ット形式
1	—	FB	80	3120	インストール用制御文	6	PO
2	—	U	—	6144	ロードモジュール	17	PO
3	—	VB	255	3120	ライブラリ	26	PO
4	—	VB	80	3120	Helpテキスト	44	PO
5	—	VB	80	3120	マニュアルテキスト	47	PO
6	—	VB	80	3120	Utilispのアセンブラ ソース	69	PS
7	—	FB	80	3120	ライブラリのUtilisp ソース	39	PO
8	—	FB	80	3120	サンプルプログラム	3	PO
9	—				データ		

注) データセットの属性は、DASD上での属性を表す。磁気テープ上では、6番目のデータセットを除いて、すべてJSECOPIYのアンロード形式(レコード形式:VS)である。9番目以降のデータセットについては、ユーザが意識する必要はない。

A. 3 UTILISPの起動と停止

システムを起動する標準的なコマンド列は以下のとおりである。

```

LOGON TSS <ユーザID> / <パスワード> SIZE(<リージョン>)
READY
LIS 'SYS1.UTILISP'
READY
UTILISP [パラメタ.....]
> :
:
:
>(QUIT)
READY

```

コマンド名はUTILISPあるいはUTIである。

リージョンは少なくとも 512 KB 程度必要である

LISPあるいはUTILISPコマンドには以下に示す省略可能なパラメタがある。

- ① STACK :STACK(n)はスタック領域の大きさをnページ (1ページ=4KB)に指定する。nの省略値は16である。
- ② FIX :FIX(n)は " 固定ヒープ " (fixed heap)領域 (コンパイルドコード局)をnページにする。nの省略値は32である。
- ③ SAVE :SAVE(n)はも入出力バッファと外面プログラムのためにnページ確保しておくように指定する。nの省略値は64である。
- ④ MANAGER :MANAGER(id)は、変数MANAGER-IDの値をストリング "(id)"にする。省略時の値はインストール時に決められる。
- ⑤ SYSPARM :SYSPARM (文字列) は、変数 SYSPARM の値をストリング "(文字列)"にする。これはユーザが自由に使ってよい。SYSPARM パラメータを省略した時は、変数 SYSPARM の値は'未束縛'である。

データ定義名 (DD名) 'LISPLIB' が割り当てられていればそのファイル (順ファイルでなければならない) が最初の実行される。この実行は結果が表示されないことを除きLispの標準トップレベル・ループと全く同じである。

LISPLIBファイル (もしあれば) の実行後、システムはトップレベル・ループに入る。読みこまれた各S式は評価 (evaluate) され、その結果が表示される。トップレ

ベルの評価関数 (evaluator) は EVALQUOTE ではなく EVAL である。

UTILISPセッションは、通常は関数QUITを呼ぶことにより終了できるが、関数ABENDにより終了することもできる。

これらのシステム関数がLISPのリーダ (reader) に認識されない場合、たとえばREADTABLEやOBJECTORが壊れた時は、端末から感嘆符「！」を連続10個以上行の始めに入力すればUTILISPセッションを終了させることができる。

終了以前あるいは予期せぬ長時間の計算が必要になった時は、端末からのアテンション割込み (普通はブレーク・キーの打鍵) でその計算を中止することができ、システムはブレーク・ループに入る。

なお、次のことに注意を要する。

関数QUITは、オープンされているストリームをすべてクローズしてからUTILISPセッションを終了させるため、エラーなどのために (オープンされたまま) 放置されていたストリームがクローズされて、それ以前にクローズされているファイルを壊すことがある。これを防ぐために、エラーが起きた時に変数OPENFILESを調べ、オープンされているストリームがあれば、その場でクローズしておくことを勧める (MAPC OPENFILES 'CLOSE)。もしクローズすることを忘れた時は、関数ABENDでUTILISPセッションを終了させる方法もある。この場合、クローズはより上位のシステムにまかされる。

スペース不足など、何らかの理由でクローズできなくなったファイルがある時はQUITできないので、関数ABENDを使ってUTILISPセッションを終わること。

表A. 2にUTILISPが保存されているファイルを示す。

表A. 2 UTILISPの保存ファイル

ファイル名	編成	内 容
J2749.UTILISP.ASM'	PS	UTILISPのアセンブラ
J2749.UTILISP.LOAD (NEWUTI)	PO	UTILISPのロードモジュール

付録B DISKETのファイル構成

診断システムDISKETのプログラムは、表B. 1に示されるようにDASD上で6つのファイルに保存されている。

表B. 1 DISKETのファイル構成

ファイル名	編成	レコード形式	レコード長	ブロック長	内 容
'J2749.DISKET.TEXT'	PO	FB	80	3120	IERIASの本体
'J2749.DISKET.NATURAL'	PO	VB	255	3120	自然言語 インターフェイス
'J2749.DISKET.GRAPH'	PO	VB	255	2560	グラフィックス
'J2749.DISKET.TOOL'	PO	FB	80	3200	知識ベース 開発支援ツール
'J2749.DISKET.TOOLIO'	PO	U	—	32760	フルスクリーン ハンドラ
'J2749.DISKET.CLIST'	PO	FB	80	3120	モード別 実行メンバー

表B. 1のファイルのメンバー一覧を表B. 3に示す。

DISKETではその他、表B. 2に示される形態のファイルをユーザーが作成して使用する。

表B. 2 DISKETで使用するファイル

ファイルの種類	編成	内 容	作成方法
知識ベース ソースファイル	P O	知識ベース記述のフォーマット に従ってかかれたソース	知識ベースエディタま たは PFD等のテキスト 編集エディタを用いる。
コンパイルされた 知識ベースファイル	P O	コンパイラにより知識ベース をS式に変換したもの	知識ベース・ソースを コンパイル・ファイル へセーブする。
ケースファイル	P O	生起事象の値と時刻	ケースジェネレータま たは PFD等のテキスト 編集エディタを用いる。

表 B. 3 メンバー一覧

ファイル名	メンバー	
'J2749.DISKET.TEXT'	CONFLG DIVCASE DIVCMD DIVCOM DIVCOMP DIVIER DIVLIST DIVLOAD DIVRUN DIVSAVE	GRAPHSET IERNAT INFER PATCH PATCHCOM PATCHSET PATCH1 PATCH2 UTI
'J2749.DISKET.NATURAL'	IEACTFN IEALL2 IECONSUL IECTM IEDCTGEN IEDICT4 IEGRAMMA	IEGRAMM IEKIHON IELOAD IEMATFN IETABLE PATCH PATCH32
'J2749.DISKET.GRAPH'	BGRAPHD D0000 ETREE FIGPATCH FIGTOOL FPATCH1 FPATCH2	GLOBAL PATCH1 PATCH2 PATCH3 PATCH4 SETUPG2

ファイル名	メンバー	
'J2749.DISKET.TOOL'	SETTOOL SETTOOL2 TLGL TOOL	
'J2749.DISKET.TOOLIO'	CLEARF TGET CLEAR TPUT STFSOFF TPUT3 STFSYES TPUT552	
'J2749.DISKET.UNIT'	ALL@N@G TOOL ALL@N INFER ALL@G RUN@N@G ALL RUN@N CASE RUN@G COMPILE RUN	

付録C 知識ベースの例

次に示す知識ベースは、SRS (Safety Research Simulator)を対象として作成したものである。44種類の事故が事故仮説として含まれている。


```

#
#   SAVED ON 87//10//12
#
#
# KNOWLEDGE BASE OF SRS
#
***DATA DEFINITION
**HYPOTHESES
*TAXONOMY
#NRA      NUCLEAR POWER PLANT ACCIDENT
FPCS      .FAILURE OF PRIMARY COOLANT SYSTEM (0.18)
FPCSPM    ..FAILURE OF PRIMARY COOLANT PUMP (0.60)
RCSPL1    ...LOSS OF REACTOR COOLANT ONE PUMP (0.30)
RCSPL2    ...LOSS OF REACTOR COOLANT TWO PUMPS (0.20)
RCSPLA    ...LOSS OF REACTOR COOLANT ALL PUMPS (0.20)
RCSPSL    ...LOSS OF REACTOR COOLANT ONE PUMP SHAFT LOCK (0.20)
FPCSLP    ..LEAK OF PRIMARY COOLANT LOOP (0.30)
RCSUTR    ...SG U-TUBE RUPTURE (0.25)
RCSLSL    ...SMALL LEAK OF REACTOR COOLANT SYSTEM (0.30)
RCRVSO    ...PRESSURIZER RELIEF VALVE STICKS OPEN (0.30)
#
FPRZ      .FAILURE OF PRESSURIZER SYSTEM (0.12)
FPRPRC    ..FAILURE OF PRESSURIZER PRESSURE CONTROL SYSTEM (0.50)
PSPVO1    ...PRESSURIZER SPRAY CONTROL ONE VALVE FAILS OPEN (0.20)
PSPVC2    ...PRESSURIZER SPARY CONTROL TWO VALVES FAIL CLOSE (0.20)
CPRPFH    ...PRESSURIZER PRESSURE CONTROL FAILS HIGH (0.25)
CPRPFL    ...PRESSURIZER PRESSURE CONTROL FAILS LOW (0.25)
FPRHTR    ..FAILURE OF PRESSURIZER HEATER (0.30)
PHTBFN    ...PRESSURIZER BACKUP HEATERS FAIL ON (0.30)
PHTPFF    ...PRESSURIZER PROPORTIONAL HEATERS FAIL OFF (0.30)
#
FCRD      .FAILURE OF CONTROL ROD SYSTEM (0.15)
FCRDCS    ..FAILURE OF ROD CONTROL SYSTEM (0.45)
CRDUCI    ...UNCONTROLLED CONTINUOUS INSERTION OF CONTROL ROD (0.20)
CRDUCW    ...UNCONTROLLED CONTINUOUS WITHDRAWAL OF CONTROL ROD (0.20)
CRDMDF    ...CONTROL FAIL TO MOVE ON DEMAND (0.20)
CRDATE    ...AUTO REACTOR TRIP FAILS (0.20)
CLATEH    ...RTD IN COLD LEG A FAILS HIGH (0.20)
FCRDDM    ..FAILURE OF ROD DRIVING MECHANISM (0.30)
CRDDR1    ...DROPPED CONTROL ROD P-6 CONTROL BANK A (0.15)
CRDEJ1    ...EJECTED ROD D-12 CONTROL BANK C (0.15)
CRDWDF    ...ROD CONTROL FAILURE DEMAND WITHDRAWAL (0.20)
CRDMAS    ...CONTROL ROD GROUPS MOVE AT MAXIMUM SPEED (0.20)
CRDMIS    ...CONTROL ROD GROUPS MOVE AT MINIMUM SPEED (0.20)
#
FCVCS     .FAILURE OF CHEMICAL AND VOLUME CONTROL SYSTEM (0.09)
FCVCLC    ..FAILURE OF VCT LEVEL CONTROL SYSTEM (0.40)
CVTLFH    ...V.C. TANK LEVEL CONTROL FAILS HIGH (0.50)
CVTLFL    ...V.C. TANK LEVEL CONTROL FAILS LOW (0.50)
FCVCCG    ..FAILURE OF CHARGING SYSTEM (0.40)
CHAGFL    ...LOSS OF CHARGING FLOW (0.30)
#
FSTM      .FAILURE OF STEAM SYSTEM (0.12)
FSTMPC    ..FAILURE OF STEAM PRESSURE CONTROL SYSTEM (0.40)
CSHPFH    ...STEAM HEADER PRESSURE CONTROL FAILS HIGH (0.30)
CSHPFL    ...STEAM HEADER PRESSURE CONTROL FAILS LOW (0.30)
CEHCAL    ...EHC CONTROL LOSS OF AUTOMATIC (0.30)
FSTMLL    ...LEAK OF STEAM LINE (0.50)
SDVLA     ...LEAKAGE OF ATOMOSPHERIC STEAM DUMP VALVE (0.10)
SGSLL     ...SG STEAM LINE LEAK (0.10)
#
FTRG      .FAILURE OF TURBINE GENERATOR SYSTEM (0.08)

```

```

FTURCS      ..FAILURE OF TURBINE CONTROL SYSTEM (0.60)
TGVFO1      ...TURBINE GOVERNOR VALVE FAILS OPEN (0.30)
TGVFC1      ...TURBINE GOVERNOR VALVE FAILS CLOSE (0.30)
TURRBF      ...FAILURE OF AUTO TURBINE RUNBACK (0.20)
TURTRP      ...TURBINE TRIP (0.20)
FGENES      ..FAILURE OF GENERATOR SYSTEM (0.20)
GENTRP      ...LOSS OF MAIN GENERATOR (0.20)
#
FCOND       .FAILURE OF CONDENSATE SYSTEM (0.09)
FCONCW      ..FAILURE OF CIRCULATING WATER SYSTEM (0.50)
CCWPL       ...PARTIAL LOSS OF CIRCULATING WATER (0.50)
CCWTL       ...TOTAL LOSS OF CIRCULATING WATER (0.40)
#
FSGFW       .FAILURE OF SG FEEDWATER SYSTEM (0.12)
FSGLCN      ..FAILURE OF SG LEVEL CONTROL SYSTEM (0.50)
CSGLFH      ...SG-B LEVEL CONTROL FAILS HIGH (0.40)
CSGLFL      ...SG-B LEVEL CONTROL FAILS LOW (0.40)
FSGFWL      ..FAILURE OF SG FEEDWATER LINE (0.40)
FWFPL       ...PARTIAL LOSS OF FEEDWATER (0.40)
FWFTL       ...TOTAL LOSS OF FEEDWATER (0.20)
HPHDPL      ...LOSS OF H.P.HEATER DRAIN PUMP (0.20)
#FRADM      .FAILURE OF RADIATION MONITER (0.10)
#
*INTERNAL HYPOTHESES
#
IHTAX1      APPLIED CONDITION OF KNOWLEDGE UNIT 2
IHTAX2      APPLIED CONDITION OF KNOWLEDGE UNIT 3
FRADM       FAILURE OF RADIATION MONITER
#
*TREATMENTS
#
TREAT01     TAKE ACTION OF TREATMENT NO.1
TREAT02     TAKE ACTION OF TREATMENT NO.2
TREAT03     TAKE ACTION OF TREATMENT NO.3
TREAT04     TAKE ACTION OF TREATMENT NO.4
TREAT05     TAKE ACTION OF TREATMENT NO.5
TREAT06     TAKE ACTION OF TREATMENT NO.6
TREAT07     TAKE ACTION OF TREATMENT NO.7
TREAT08     TAKE ACTION OF TREATMENT NO.8
TREAT09     TAKE ACTION OF TREATMENT NO.9
TREAT10     TAKE ACTION OF TREATMENT NO.10
TREAT11     TAKE ACTION OF TREATMENT NO.11
TREAT12     TAKE ACTION OF TREATMENT NO.12
TREAT13     TAKE ACTION OF TREATMENT NO.13
TREAT14     TAKE ACTION OF TREATMENT NO.14
TREAT15     TAKE ACTION OF TREATMENT NO.15
TREAT16     TAKE ACTION OF TREATMENT NO.16
TREAT17     TAKE ACTION OF TREATMENT NO.17
TREAT18     TAKE ACTION OF TREATMENT NO.18
TREAT19     TAKE ACTION OF TREATMENT NO.19
TREAT20     TAKE ACTION OF TREATMENT NO.20
TREAT21     TAKE ACTION OF TREATMENT NO.21
TREAT22     TAKE ACTION OF TREATMENT NO.22
TREAT23     TAKE ACTION OF TREATMENT NO.23
TREAT24     TAKE ACTION OF TREATMENT NO.24
TREAT25     TAKE ACTION OF TREATMENT NO.25
TREAT26     TAKE ACTION OF TREATMENT NO.26
TREAT27     TAKE ACTION OF TREATMENT NO.27
TREAT28     TAKE ACTION OF TREATMENT NO.28
TREAT29     TAKE ACTION OF TREATMENT NO.29
TREAT30     TAKE ACTION OF TREATMENT NO.30
TREAT31     TAKE ACTION OF TREATMENT NO.31

```

TREAT32 TAKE ACTION OF TREATMENT NO.32
 TREAT33 TAKE ACTION OF TREATMENT NO.33
 TREAT34 TAKE ACTION OF TREATMENT NO.34
 TREAT35 TAKE ACTION OF TREATMENT NO.35
 TREAT36 TAKE ACTION OF TREATMENT NO.36
 TREAT37 TAKE ACTION OF TREATMENT NO.37
 TREAT38 TAKE ACTION OF TREATMENT NO.38
 TREAT39 TAKE ACTION OF TREATMENT NO.39
 TREAT40 TAKE ACTION OF TREATMENT NO.40
 TREAT41 TAKE ACTION OF TREATMENT NO.41
 TREAT42 TAKE ACTION OF TREATMENT NO.42
 TREAT43 TAKE ACTION OF TREATMENT NO.43

#

#

**FINDINGS

*BEGIN QUESTIONNAIRE

#

*MULTIPLE CHOICE

TYPE OF PROBLEM:

NPPA NUCLEAR POWER PLANT ACCIDENT

OTHA OTHER ACCIDENT

#

ANNUNCIATORS

#

*CHECKLIST

ANNUNCIATORS OF TRIP:

ATRPRE REACTOR TRIP

ATRPTR TURBINE TRIP

ATRPGE GENERATOR TRIP

*CHECKLIST (2)

ANNUNCIATORS OF CONTROL ROD:

ACRDSTP ROD STOP

ACRDBTM ROD BOTTOM

ACRDWIL ROD WITHDRAW LIMIT

#

*CHECKLIST (5)

ANNUNCIATORS OF PRESSURIZER:

ALEVPRH PRESSURIZER LEVEL HIGH

ALEVPRL PRESSURIZER LEVEL LOW

APREPRH PRESSURIZER PRESSURE HIGH

APREPRL PRESSURIZER PRESSURE LOW

ATEPRLH PRESSURIZER RELIEF LINE HIGH TEMP.

ATEPRTH PRESSURIZER RELIEF TANK HIGH TEMP.

APRERTH PRESSURIZER RELIEF TANK HIGH PRESS.

#

*CHECKLIST (5)

ANNUNCIATORS OF PRIMARY LOOP:

AFLRCSL LOW REACTOR COOLANT FLOW

ATEAVRF DEVIATION OF TAVG/TREF

ATEAVGH AVERAGE TEMPERATURE HIGH

ATEAVGL AVERAGE TEMPERATURE LOW

*CHECKLIST (5)

ANNUNCIATORS OF CVCS:

APREVTH VOLUME CONTROL TANK HIGH PRESS.

APREVTL VOLUME CONTROL TANK LOW PRESS.

ATEMVTH VOLUME CONTROL TANK HIGH TEMP.

ATEMLDH LETDOWN FLOW HIGH TEMP.

AFLCHAH CHARGING FLOW HIGH

AFLCHAL CHARGING FLOW LOW

ALEVRWL REFUEL WATER STORAGE TANK LOW LEVEL

#

*CHECKLIST (5)

ANNUNCIATORS OF SG AND STEAM LINE:

ALESGE STEAM GENERATOR LEVEL ERROR
 ALESGBH STEAM GENERATOR B ALARM HIGH LEVEL
 ALESGBL STEAM GENERATOR B ALARM LOW LEVEL
 ALESGCH STEAM GENERATOR C ALARM HIGH LEVEL
 ALESGCL STEAM GENERATOR C ALARM LOW LEVEL
 AFSLH STEAM LINE FLOW HIGH
 APRSL STEAM LINE PRESSURE LOW
 AVISOLC STEAM LINE ISOLATION
 AVSDVAR STEAM DUMP ARMED

#

*CHECKLIST (5)

ANNUNCIATORS OF FEEDWATER SYSTEM:

AVISOWF FEEDWATER ISOLATION
 AHTRWFT FEEDWATER HEATER TROUBLE
 APRECVL CONDENSER VACCUM TROUBLE
 APRECON CONTAINMENT PARTIAL PRESSURE HIGH
 ALPHDH HIGH PRESSURE HEATER DRAIN TANK LEVEL HIGH
 ALPHDL HIGH PRESSURE HEATER DRAIN TANK LEVEL LOW

#

*CHECKLIST (2)

ANNUNCIATORS OF TURBINE GENERATOR SYSTEM:

ARUNBTR TURBINE RUNBACK
 ASPTURL TURBINE ZERO SPEED
 AOFTDRL AUTO STOP OIL TO DRAIN
 AVOLRGT VOLTAGE REGULATOR TROUBLE
 ATRPEXF EXCITER FIELD BREAKER TRIP
 ATRPBR4 4KV BREAKER AUTO TRIP

#

*CHECKLIST (3)

OTHER ANNUNCIATORS:

ARADMAL RADIATION MONITOR ALARM
 AINISI SAFETY INJECTION INITIATION
 ABLCSI SAFETY INJECTION BLOCK
 AINISG CONSEQUENCE LIMITING SAFEGUARD INITIATION
 #ASFLUX HIGH FLUX AT SHUTDOWN

#

#

REACTOR TRIP CONDITIONS

*CHECKLIST

SCRAM CONDITIONS:

SCRAM REACTOR SCRAM
 SRCSL1 RCS LOW FLOW 1/3
 SRCSL2 RCS LOW FLOW 2/3
 SPRFXH POWER RANGE FLUX HIGH
 STURTR TURBINE TRIP
 SPRZPL PRESSURIZER LOW PRESSURE
 SPRZPH PRESSURIZER HIGH PRESSURE
 SPRZLH PRESSURIZER HIGH LEVEL
 SOVRTE OVER TEMPERATURE
 SOVRPO OVER POWER
 SSGLEL SG LOW LEVEL
 SSISIG SI SIGNAL ON
 # SSRFXH SOURCE RANGE FLUX HIGH
 # SIRRTH INTERMEDIATE RANGE STARTUP RATE HIGH

#

TURBINE TRIP CONDITIONS

*CHECKLIST

TURBINE TRIP:

TURTRIP TURBINE TRIP
 TURTST TURBINE TRIP SOLENOID TROUBLE
 TURCPH CONDENSER HIGH PRESSURE

TURSP0 TURBINE OVER SPEED
 TURLB0F TURBINE LATCH BUTTON OFF
 TURTBO TURBINE TRIP BUTTON ON
 #
 # ENGINEERED SAFEGUARDS SYSTEM ACTUATION CONDITIONS
 *CHECKLIST
 ESS SIGNSL:
 ESSACT ESS SIGNAL ACTUATION
 ESCOPH CONTAINMENT PRESSURE HIGH
 ESPRL PRESSURIZER BOTH LEVEL AND PRESS. LOW
 ESSGHP DIFFER.PRESS. SG AND S.H HIGH
 ESSTFH STEAM FLOW HIGH
 ESSLPL STEAM LINE PRESSURE LOW
 ESTAVL AVERAGE TEMPERATURE LOW
 #
 # NUMERICAL
 #
 *NUMERICAL (0 150) (5 I)
 FLRCS1 REACTOR COOLANT FLOW NO.1 LOOP (%)
 *NUMERICAL (0 150) (5 I)
 FLRCS2 REACTOR COOLANT FLOW NO.2 LOOP (%)
 *NUMERICAL (0 150) (5 I)
 FLRCS3 REACTOR COOLANT FLOW NO.3 LOOP (%)
 *NUMERICAL (7 I)
 GFRCS1 GRADIENT OF COOLANT FLOW NO.1 LOOP (%)
 *NUMERICAL (7 I)
 GFRCS2 GRADIENT OF COOLANT FLOW NO.2 LOOP (%)
 *NUMERICAL (7 I)
 GFRCS3 GRADIENT OF COOLANT FLOW NO.3 LOOP (%)
 *NUMERICAL (0 250) (5 I)
 PREPRZ PRESSURIZER PRESSURE (KG CM2.G)
 *NUMERICAL (0 100) (5 I)
 LEVPRZ PRESSURIZER LEVEL (%)
 *NUMERICAL (7 I)
 LEVPRT DEVIATION OF TAVG AND PRESSURIZER LEVEL (-)
 *NUMERICAL (7 I)
 TEMPRL PRESSURIZER RELIEF LINE TEMPERATURE (DEG.C)
 *NUMERICAL (7 I)
 TEMPRT PRESSURIZER RELIEF TANK TEMPERATURE (DEG.C)
 *NUMERICAL (7 I)
 PREPRT PRESSURIZER RELIEF TANK PRESSURE (KG CM2.G)
 *NUMERICAL (7 I)
 TEMHL1 HOT LEG TEMPERATURE NO.1 LOOP (DEG.C)
 *NUMERICAL (7 I)
 TEMHL2 HOT LEG TEMPERATURE NO.2 LOOP (DEG.C)
 *NUMERICAL (7 I)
 TEMHL3 HOT LEG TEMPERATURE NO.3 LOOP (DEG.C)
 *NUMERICAL (7 I)
 TEMCL1 COLD LEG TEMPERATURE NO.1 LOOP (DEG.C)
 *NUMERICAL (7 I)
 TEMCL2 COLD LEG TEMPERATURE NO.2 LOOP (DEG.C)
 *NUMERICAL (7 I)
 TEMCL3 COLD LEG TEMPERATURE NO.3 LOOP (DEG.C)
 *NUMERICAL (7 I)
 TEMAVE AVERAGE TEMPERATURE (DEG.C)
 *NUMERICAL (7 I)
 TEMDEV DEVIATION OF TAVG AND TREF (-)
 *NUMERICAL (7 I)
 QFLUX AVERAGE NEUTRON FLUX (%)
 *NUMERICAL (0 220) (7 I)
 LCRODA CONTROL ROD POSITION GROUP A (STEP)
 *NUMERICAL (0 220) (5 I)

```

LCRODB CONTROL ROD POSITION GROUP B (STEP)
*NUMERICAL (0 220) (5 I)
LCRODC CONTROL ROD POSITION GROUP C (STEP)
*NUMERICAL (0 220) (5 I)
LCRODD CONTROL ROD POSITION GROUP D (STEP)
*NUMERICAL (0 80) (5 I)
SPCROD CONTROL ROD SPEED (STEP MIN)
*NUMERICAL (5 I)
FLCHAG CHARGING FLOW (TON HR)
*NUMERICAL (7 I)
TEMLD LET DOWN FLOW TEMPERATURE (DEG.C)
*NUMERICAL (7 I)
PREVCT VOLUME CONTROL TANK PRESSURE (KG CM2.A)
*NUMERICAL (7 I)
TEMVCT VOLUME CONTROL TANK TEMPERATURE (DEG.C)
#
# SECONDARY SYSTEM
*NUMERICAL (0 100) (7 I)
LEVSG1 SG LEVEL NO.1 (%)
*NUMERICAL (0 100) (7 I)
LEVSG2 SG LEVEL NO.2 (%)
*NUMERICAL (0 100) (7 I)
LEVSG3 SG LEVEL NO.3 (%)
*NUMERICAL (7 I)
PRESG1 SG STEAM PRESSURE NO.1 (KG/CM2.G)
*NUMERICAL (7 I)
PRESG2 SG STEAM PRESSURE NO.2 (KG/CM2.G)
*NUMERICAL (7 I)
PRESG3 SG STEAM PRESSURE NO.3 (KG/CM2.G)
*NUMERICAL (7 I)
FLMFW1 MAIN FEEDWATER FLOW NO.1 (%)
*NUMERICAL (7 I)
FLMFW2 MAIN FEEDWATER FLOW NO.2 (%)
*NUMERICAL (7 I)
FLMFW3 MAIN FEEDWATER FLOW NO.3 (%)
*NUMERICAL (7 I)
TEMPFW MAIN FEEDWATER TEMPERATURE (DEG.C)
*NUMERICAL (7 I)
FLMSF1 MAIN STEAM FLOW NO.1 (%)
*NUMERICAL (7 I)
FLMSF2 MAIN STEAM FLOW NO.2 (%)
*NUMERICAL (7 I)
FLMSF3 MAIN STEAM FLOW NO.3 (%)
*NUMERICAL (7 I)
LESGLP SG LEVEL DEVIATION (-)
*NUMERICAL (7 I)
PRETR1 TURBINE FIRST STAGE PRESSURE (KG/CM2.G)
*NUMERICAL (7 I)
QGENETR GENERATOR POWER (MWE)
*NUMERICAL (0 10) (7 I)
PRCOND CONDENSER PRESSURE (KG/CM2.A)
*NUMERICAL (0 10) (7 I)
PRCONT CONTAINMENT PRESSURE (KG/CM2.A)
*NUMERICAL (7 I)
TECONT CONTAINMENT TEMPERATURE (DEG.C)
*NUMERICAL (7 I)
LERWST REFUEL WATER STORAGE TANK LEVEL (M)
*NUMERICAL (7 I)
NUMTUR TURBINE ROTATION NUMBER (RPM)
#
# PUMP
*CHECKLIST (3)

```

PUMPS:		
PMRCP1	REACTOR COOLANT PUMP 1A	RCP-1A
PMRCP2	REACTOR COOLANT PUMP 2A	RCP-2A
PMRCP3	REACTOR COOLANT PUMP 3A	RCP-3A
PMCHAG	CHARGING PUMP	1-CH-P1A
PMRHR	RHR PUMP	RH-P-1A
PMSGFW	SG FEEDWATER PUMP	FW-P-1B
PMAFW	SG AUX. FEEDWATER PUMP	FW-P-3A
PMCIRA	CIRCULATING WATER PUMP	CW-P-1A
PMCIRB	CIRCULATING WATER PUMP	CW-P-1B
PMCIRS	EMERGENCY SERVICE WATER PUMP	SW-P-1A
PMCOND	CONDENSER PUMP	CN-P-1A
PMCNSP	CONTAINMENT SPRAY PUMP	1-CS-P1A
PMSIS	SIS PUMP	1-SI-P1A
PMVACU	VACUUM PUMP	1-VP-P2A
PMHPHD	HP HEATER DRAIN PUMP	SD-P-1A
PMLPHD	LP HEATER DRAIN PUMP	SD-P-2A
PMBOA	BORIC ACID TRANSFER PUMP	1-CH-P2A
PMRCIR	INSIDE RECIRCULATION PUMP	RS-P-1A
PMPWST	PWST PUMP	PG-P-1

*CHECKLIST (5)

HEATER:
HTRBUP BACK UP HEATER ON
HTRPRP PROPORTIONAL HEATER ON

VALVE

OPEN...TRUE CLOSE...FALSE
#

*CHECKLIST (5)
VALVE POSITION:

VP455A	PRESSURIZER SPRAY VALVE A	PCV 455A
VP455B	PRESSURIZER SPRAY VALVE C	PCV 455B
VF605	RHR PUMP EXIT VALVE (OR RHR BYPASS VALVE)	FCV 605
VF1122	VALVE FROM CHARGING PUMP TO RHE	FCV 1122
VP1145	NONREGENERATION H.E EXIT VALVE	PCV 1145
VF1113A	BLENDER INLET VALVE FROM BAT PUMP	FCV 1113A
VF1114A	BLENDER INLET VALVE FROM PWST PUMP	FCV 1114A
VCH218	BORIC ACID DILUTION VALVE	1 CH 218
VH1244	DEBORATION DEMINERATION DIVERT VALVE	HCV 1244
VRC139	THROTTLE VALVE OF SPRAY LINE A	RC 139
VRC140	THROTTLE VALVE OF SPRAY LINE C	RC 140
VM100		MOV 100
VM1350	VALVE FROM BORIC ACID TANK TO CHAR. PUMP	MOV 1350
VP455C	PRESSURIZER RELIEF VALVE	PCV 455C
VM563	PRESSURIZER RELIEF MOTOR VALVE	MOV 563
VPRSV	PRESSURIZER SAFETY VALVE	SV 551C
V1115A	DIVERT VALVE TO VCT AND BORON RECOVERY	LCV 1115A
V1115B	RWST EXIT VALVE TO CHARGING LINE	LCV 1115B
V1115C	VCT EXIT VALVE	LCV 1115C
VM700	RHR LINE VALVE FROM HOT LEG TO RHR PUMP	MOV 700
VM867A		MOV 867A
VH758	RHR EXIT VALVE	HCV 758
VH1142	RHR ISOLATION VALVE	HCV 1142
VM1289A	VALVE FROM CHARGING PUMP TO FCV-1122	MOV 1289A
VT1143	DIVERT VALVE TO DEBORATOR AND LCV-1115A	TCV 1143
VL1460A	LET DOWN LINE VALVE	LCV 1460A
#		
VM862	RWST EXIT VALVE TO LP SI PUMP	MOV 862
VM863A	VALVE FROM LP SI PUMP TO CHARGING PUMP	MOV 863A
VM860A	VALVE FROM CONTAINMENT TO LP SI PUMP	MOV 860A

VM890A	LP SI LINE VALVE TO HOT LEG	MOV 890A
VM890C	LP SI LINE VALVE TO COLD LEG	MOV 890C
VM869B	CHARGING LINE	MOV 869B
VM720A	VALVE FROM ACCUMLATOR TANK A EXIT AND RHR	MOV 720A
VM720B	VALVE FROM ACCUMLATOR TANK B EXIT AND RHR	MOV 720B
VM865A	ACCUMLATOR TANK A EXIT VALVE	MOV 865A
VM865B	ACCUMLATOR TANK B EXIT VALVE	MOV 865B
VM865C	ACCUMLATOR TANK C EXIT VALVE	MOV 865C
#		
#	STEAM LINE VALVE	
	*CHECKLIST (10)	
	VALVE POSITION OF SECONDARY SYSTEM:	
VT105A	STEAM DUMP VALVE TO MAIN CONDENSER	TCV 105A
VH104	STEAM DUMP VALVE TO ATMOSPHERE	HCV MS 104
VMS84	STEAM LINE A BYPASS VALVE	1 MS 84
VMS116	STEAM LINE B BYPASS VALVE	1 MS 116
VMS155	STEAM LINE C BYPASS VALVE	1 MS 155
VT101A	MAIN STEAM ISOLATION VALVE A	TV MS 101A
VT101B	MAIN STEAM ISOLATION VALVE B	TV MS 101B
VT101C	MAIN STEAM ISOLATION VALVE C	TV MS 101C
VN101B	NON RETURN VALVE OF STEAM LINE B	NRV MS 101B
VN101C	NON RETURN VALVE OF STEAM LINE C	NRV MS 101C
VTMSV1	TURBINE MAIN STOP VALVE 1	TV 1
VTMSV2	TURBINE MAIN STOP VALVE 2	TV 2
VTMSV3	TURBINE MAIN STOP VALVE 3	TV 3
VTMSV4	TURBINE MAIN STOP VALVE 4	TV 4
VTGV1	TURBINE GOVERNOR VALVE 1	TGV 1
VTGV2	TURBINE GOVERNOR VALVE 2	TGV 2
VTGV3	TURBINE GOVERNOR VALVE 3	TGV 3
VTGV4	TURBINE GOVERNOR VALVE 4	TGV 4
VINTC	INTERCEPTOR VALVE	TINTCEPT
VSAFE1	S.L. SAFETY VALVE 1	SV 106A
VSAFE2	S.L. SAFETY VALVE 2	SV 107A
VSAFE3	S.L. SAFETY VALVE 3	SV 108A
VSAFE4	S.L. SAFETY VALVE 4	SV 109A
VSAFE5	S.L. SAFETY VALVE 5	SV 110A
VSAFE6	S.L. SAFETY VALVE 6	SV 111A
#		
#	FEEDWATER LINE VALVE	
VF478	FEEDWATER CONTROL VALVE A (MAIN)	FCV 478
VF488	FEEDWATER CONTROL VALVE B (MAIN)	FCV 488
VF498	FEEDWATER CONTROL VALVE C (MAIN)	FCV 498
VM154A	FEEDWATER VALVE A (MOTOR)	MOV FW 154A
VM154B	FEEDWATER VALVE B (MOTOR)	MOV FW 154B
VM154C	FEEDWATER VALVE C (MOTOR)	MOV FW 154C
VH155A	FEEDWATER CONTROL VALVE A (RESERVE)	HCV FW 155A
VH155B	FEEDWATER CONTROL VALVE B (RESERVE)	HCV FW 155B
VH155C	FEEDWATER CONTROL VALVE C (RESERVE)	HCV FW 155C
VM151A	AUX. FEEDWATER VALVE TO SG A	MOV FW 151A
VM151B	AUX. FEEDWATER VALVE TO SG B	MOV FW 151B
VM151C	AUX. FEEDWATER VALVE TO SG C	MOV FW 151C
#		
VL102A	CONDENSATE STORAGE TANK EXIT VALVE TO CO.	LCV CN 102A
VL101	CONDENSATE STORAGE TANK INLET VALVE	LCV CN 101
VF104A	RE.H.E MOIS. SEP. INLET VALVE	FCV 104A
VL082	MOIS. SEP. DR. T EXIT VALVE TO HP DR. T	LCV SD 082
VL106	HP DR. T PUMP EXIT VALVE TO FW PUMP	LCV SD 106
VL100A	RE. DR. T EXIT VALVE	LCV SD 100A
VL122A	LP HEATER DRAIN PUMP EXIT VALVE	LCV SD 122A
VL081	HP HEATER 2 EXIT VALVE	LCV SD 081
VL103B	HP HEATER 1 EXIT VALVE	LCV SD 103B
VL124A	LP HEATER 1 EXIT VALVE TO CONDENSER	LCV SD 124A

VT109A	VALVE TO EXHAUST HOOD SPRAY	TCV CN 109A
VF100	VALVE FROM HP DR PUMP TO CONDENSER	FCV SD 100
VN101	VALVE FROM HP TURBINE TO HP HEATER 1	NRV 101
VN102	VALVE FROM HP TURBINE TO HP HEATER 2	NRV 102
VL107	VALVE FROM HP DR TANK TO CONDENSER	LCV SD 107
VL101A	VALVE FROM RE HTEAR DR TANK 1A TO COND	LCV SD 101A
VL123A	VALVE FROM 4TH HTR TO CONDENSER	LCV SD 123A
VF107	VALVE FROM M.COND TO FW	FCV CN 107
VL104B		LCV SD 104B
VL120A	VALVE FROM HEATER 3 TO 4	LCV SD 120A
VL121A	VALVE FROM HEATER 3 TO CONDENSER	LCV SD 121A
VL109A	VALVE FROM DRAIN COOLER TO CONDENSER	LCV SD 127A
VL132A	VALVE FROM HEATER 2 TO CONDENSER	CV SD 132A
VCN124	HEATERS 5 AND 6 CONDENSATE BLOCK VALVE	CN 124
VCN125	HEATERS 3 AND 4 CONDENSATE INLET VALVE	CN 124
VCN126	HEATERS 3 AND 4 CONDENSATE BYPASS VALVE	CN 124
VCN132	HEATERS 3 AND 4 CONDENSATE OUTLET VALVE	CN 124
VCN195	HEATERS 5 AND 6 CONDENSATE BYPASS VALVE	CN 124
VFW107	HEATER 1 FEEDWATER INLET VALVE	FW 107
VFW108	HEATER 1 FEEDWATER BYPASS VALVE	FW 108
VF101A	VALVE FROM ATM TO CONDENSER	FCV VP 101A
VAS83	STEAM SUPPLY TO CONT AIR EJECTOR	AS 83
VH100	CONTAINMENT EVACUATE CONTROL VALVE	HCV CV 100
VCV1	CONTAINMENT EVACUATION ISOLATION	CV 1
VM100A	VALVE FROM RWST TO CONTAINMENT SPRAY PUMP	MOV CS 100A
VS100A	SG A BLOW DOWN VALVE	VSBD 100A
VS100B	SG B BLOW DOWN VALVE	VSBD 100B
VS100C	SG C BLOW DOWN VALVE	VSBD 100C

#

YES/NO

*YESNO (5)

YREVFL1 REVERSE FLOW LOOP 1

*YESNO (5)

YREVFL2 REVERSE FLOW LOOP 2

*YESNO (5)

YREVFL3 REVERSE FLOW LOOP 3

*YESNO (5)

YTEMCH1 COLD LEG TEMPERATURE HIGHER THAN HOT LEG LOOP 1

*YESNO (5)

YTEMCH2 COLD LEG TEMPERATURE HIGHER THAN HOT LEG LOOP 2

*YESNO (5)

YTEMCH3 COLD LEG TEMPERATURE HIGHER THAN HOT LEG LOOP 3

#

*END QUATIONNAIRE

#

****ACTION DEFINITION

***ACTION CASE-LOAD

**LANGUAGE L I

#(DEFUN CASE-LOAD NIL

(CALL 'ALLOC "DA(HF.DATA) F(FT02F001)"")

(CALL 'ALLOC "DA(IERIAS.DATA) F(FT01F001)"")

(CALL 'ALLOC "DA(IECASEF.DATA) F(FT03F001)"")

(CALL 'CALL "CASEMAKE.LOAD(TEMPNAME)"")

(CALL 'FREE "F(FT01F001)"")

(CALL 'FREE "F(FT02F001)"")

(CALL 'FREE "F(FT03F001)"")

(EXFILE "IECASEF.DATA")

#(D*IE-TABLE NIL))

##(CASE-LOAD)

***GENERAL KB

**KU

*FF RULES

NUMERICAL & ANN.

```

#
F(PRCONT ,#, 0.62 : * ) -> F(AINISG , T )
F(LCRODD ,#, 0 : 10 ) -> F(ACRDBTM , T )
F(LCRODD ,#, 220 : * ) -> F(ACRDWIL , T )
F(LEVPRT ,#, 0.05 : * ) -> F(ALEVPRH , T )
F(LEVPRT ,#, -1: -0.06) -> F(ALEVPRL , T )
F(PREPRZ ,#, 163 : * ) -> F(APREPRH , T )
F(PREPRZ ,#, 0 : 154 ) -> F(APREPRL , T )
F(TEMPRL ,#, 77 : * ) -> F(ATEPRLH , T )
F(TEMPRT ,#, 52 : * ) -> F(ATEPRTH , T )
F(PREPRT ,#, 0.2 : * ) -> F(APRERTH , T )
F(LERWST ,#, 0 : 13 ) -> F(ALEVRWL , T )
F(FLRCS1 ,#, 0 : 92 ) -> F(AFLRCSL , T )
F(PREVCT ,#, 3.6 : * ) -> F(APREVTH , T )
F(PREVCT ,#, 0 : 0.7 ) -> F(APREVTL , T )
F(TEMVCT ,#, 79 : * ) -> F(ATEMVTH , T )
F(TEMLED ,#, 293 : * ) -> F(ATEMLDH , T )
F(FLCHAG ,#, 25 : * ) -> F(AFLCHAH , T )
F(FLCHAG ,#, 0 : 7 ) -> F(AFLCHAL , T )
F(TEMDEV ,#, 0.05 : * ) -> F(ATEAVRF , T )
F(TEMAVE ,#, 305 : * ) -> F(ATEAVGH , T )
F(TEMAVE ,#, 0 : 286 ) -> F(ATEAVGL , T )
F(LESGLP ,#, 0.1 : * ) -> F(ALESGE , T )
F(LEVSG2 ,#, 75 : * ) -> F(ALESGBH , T )
F(LEVSG2 ,#, 0 : 18 ) -> F(ALESGBL , T )
F(LEVSG3 ,#, 75 : * ) -> F(ALESGCH , T )
F(LEVSG3 ,#, 0 : 18 ) -> F(ALESGCL , T )
F(FLMSF1 ,#, 2040 : * ) -> F(AFLSLH , T )
F(PRESG1 ,#, 0 : 36 ) -> F(APRSLL , T )
F(PRETR1 ,#, * : 10 ) -> F(AVSDVAR , T )
F(LEVSG1 ,#, 65 : * ) -> F(AVISOWF , T )
F(PRCOND ,#, 0.37 : * ) -> F(APRECVL , T )
F(PRCONT ,#, 0.71 : * ) -> F(APRECON , T )
F(NUMTUR ,#, 0 : 750 ) -> F(ASPTURL , T )

```

NUMERICAL & SCRAM COND

```

#
(2 : F(LEVSG1, #, 0:13), F(LEVSG2, #, 0:13), F(LEVSG3, #, 0:13)) -> F(SSGLEL, T)
(1 : F(FLRCS1, #, 0:92), F(FLRCS2, #, 0:92), F(FLRCS3, #, 0:92)) &
F(QFLUX, 30:*) -> F(SRCSL1, T)
(2 : F(FLRCS1, #, 0:92), F(FLRCS2, #, 0:92), F(FLRCS3, #, 0:92)) &
F(QFLUX, 10:*) -> F(SRCSL2, T)
F(QFLUX , 108 : * ) -> F(SPRFXH , T )
F(PREPRZ ,#, 0 : 130 ) & F(QFLUX ,#, 10 : * ) -> F(SPRZPL , T )
F(PREPRZ ,#, 165 : * ) & F(QFLUX ,#, 10 : * ) -> F(SPRZPH , T )
F(LEVPRT ,#, 90 : * ) & F(QFLUX ,#, 10 : * ) -> F(SPRZLH , T )
#
F(PMSGFW ,#, F ) -> F(TURTST , T )
F(STURTR , T ) -> F(ATRPTR , T )
F(ATRPTR , T ) -> F(TURTRIP , T )
#F(TURTRIP , T ) -> F(VT105A , T )
#F(ESSACT , #, T ) -> F(AINISG , T )
F(ESSACT , T ) -> F(AINISG , T )
#F(ESSACT , #, T ) -> F(PMAXFW , T )
F(ESSACT , T ) -> F(PMAXFW , T )
F(ESSACT , T ) -> F(SCRAM , T )
F(SCRAM , T ) -> F(ACRDBTM , T )
F(VP455C , T ) -> F(ATEPRLH , T )
(1 : F(ATRPTR , T), F(TURTRIP, T)) -> F(SCRAM , T )
(1 : F(SRCSL1, T), F(SRCSL2, T), F(SPRFXH, T), F(STURTR, T), F(SPRZPL, T),
F(SPRZPH, T), F(SPRZLH, T), F(SSGLEL, T), F(SSISIG, T)) -> F(SCRAM, T)

```

```

(1 : F(TURTST,T),F(TURCPH,T),F(TURSPQ,T),F(TURLBF,T),F(TURTBO,T))
-> F(TURTRIP,T)
#
#
*FH RULE
#
# NO.1 FPCS
#
#F(VL1460A,T) & F(FLCHAG,20:*) & F(ARADMAL,T) -> H(FRADM , 0.50 )
#F(NPPA,T) -> H(FRADM , 0.00 )
F(FLCHAG,#,15:*) & F(ARADMAL,T) -> H(FRADM , 0.05)
F(SPRZPL , T ) -> H(FPCS , 0.05 )
(1 : F(SRCSL1,T),F(SRCSL2,T)) -> H(FPCS , 0.50 )
F(AFLRCSL,T) -> H(FPCS , 0.30 )
(1 : F(FLRCS1,-50:50),F(FLRCS2,-50:50),F(FLRCS3,-50:50))
-> H(FPCS, 0.50)
F(APRECON,T) -> H(FPCS , 0.45 )
(1 : F(AFLCHAH,T),F(FLCHAG, 20:*)) -> H(FPCS , 0.35 )
F(ARADMAL,T) -> H(FPCS , 0.60 )
F(ESSACT ,T) -> H(FPCS , 0.40 )
F(ALEVPRL,T) & F(APREPRL,T) -> H(FPCS , 0.30 )
(1 : F(PRCNT, 0.65:*),F(TECONT,50:*)) -> H(FPCS , 0.40 )
(1 : F(ATEPRLH, T),F(ATEPRTH, T),F(APRERTH, T)) -> H(FPCS , 0.15)
#
# NO.2 FPRZ
#
F(SPRZPL , T ) -> H(FPRZ , 0.40)
F(SSGLEL , T ) -> H(FPRZ , 0.10)
F(ATEPRLH, T ) -> H(FPRZ , 0.20)
(1 : F(APREPRH, T),F(APREPRL, T)) -> H(FPRZ , 0.15)
(1 : F(ALEVPRL, T),F(ALEVPRH, T)) -> H(FPRZ , 0.20)
(1 : F(ATEPRLH, T),F(ATEPRTH, T),F(APRERTH, T)) -> H(FPRZ , 0.15)
#
# NO.3 FCRD
#
F(SPRFXH , T ) -> H(FCRD , 0.30)
F(ACRDWIL,#, T ) -> H(FCRD , 0.30)
F(ATEAVRF,#, T ) -> H(FCRD , 0.05)
F(ACRDSTP,#, T ) -> H(FCRD , 0.25)
F(ACRDBTM,#, T ) -> H(FCRD , 0.05)
#
# NO.4 FCVCS
#
(1 : F(AFLCHAH, T),F(AFLCHAL , T )) -> H(FCVCS , 0.30)
(1 : F(APREVTL, T),F(APREVTH , T )) -> H(FCVCS , 0.30)
F(VL1460A ,#,F) -> H(FCVCS , 0.20)
F(V1115B , T ) -> H(FCVCS , 0.20)
F(PMCHAG ,#, F ) -> H(FCVCS , 0.30)
#
# NO.5 FSTM
#
F(SSGLEL ,#, T ) -> H(FSTM , 0.10)
F(APRSL ,#, T ) -> H(FSTM , 0.30)
F(AFLSLH ,#, T ) -> H(FSTM , 0.10)
F(VH104 , T ) -> H(FSTM , 0.10)
F(PMAXFW,#,F) -> H(FSTM , 0.10)
F(ESSACT , T ) -> H(FSTM , 0.20)
#
# NO.6 FTRG
#
F(STURTR , T ) -> H(FTRG , 0.40)
(1 : F(AVOLRGT, T),F(ATRPEXF, T)) -> H(FTRG , 0.20)

```

```

(1 : F(ATRPTR , T),F(ATRPGE , T)) -> H(FTRG , 0.20)
F(ARUNBTR, T ) -> H(FTRG , 0.20)
F(ASPTURL, T ) -> H(FTRG , 0.30)
#
# NO.7 FCOND
#
F(STURTR , T ) -> H(FCOND , 0.30)
F(APRECVL, T ) -> H(FCOND , 0.50)
F(TURCPH , T ) -> H(FCOND , 0.50)
(1 : F(PMCIRA,F),F(PMCIRB,F)) -> H(FCOND , 0.60)
#
# NO.8 FSGFW
#
(1 : F(SSGLEL,T),F(TURTRIP,T)) & F(PMAXFW,T) -> H(FSGFW , 0.40)
F(SPRFXH , T ) -> H(FSGFW , 0.10)
F(ALEVPRH, T ) -> H(FSGFW , 0.20)
(1 : F(ALESGBH,T),F(ALESGBL,T)) -> H(FSGFW , 0.30)
(1 : F(ALESGCH,T),F(ALESGCL,T)) -> H(FSGFW , 0.30)
#
# **** TAX1
*HH RULE
#
*IF
F(NPPA,T)
*THEN
(1 : H(FPCS, 0.1: 1.0),H(FPRZ,0.1:1.0)) -> H(IHTAX1, 0.4)
(1 : H(FCRD, 0.1: 1.0),H(FCVCS,0.1:1.0)) -> H(IHTAX1, 0.4)
(1 : H(FSTM, 0.1: 1.0),H(FTRG ,0.1:1.0)) -> H(IHTAX1, 0.4)
(1 : H(FCOND , 0.1: 1.0),H(FSGFW , 0.1:1.0)) -> H(IHTAX1, 0.4)
*END
#SECOND GRADE OF TAXONOMY
#
**KU
*COND H(IHTAX1, 0.4: 1.0)
#
*HH RULES
#
# NO.1 FPCSPM,FPCSLP
#
*IF
H(FPCS , 0.1:1.0)
*THEN
(1 : F(FLRCS1,0:90),F(FLRCS2,0:90),F(FLRCS3,0:90),F(AFLRCSL,T))
-> H(FPCSPM,0.6)
(1 : F(AFLCHAH,T),F(FLCHAG,20: *),F(APREVTL,T),F(VL1460A,F))
-> H(FPCSLP,0.5)
F(AFLRCSL , F ) -> H(FPCSPM , -0.50)
F(ACRDWIL , T ) -> H(FPCSLP , 0.20)
F(ALEVPRL , T ) & F(APREPRL,T) -> H(FPCSLP , 0.30)
F(PMCNSP , T ) -> H(FPCSLP , 0.50)
F(ESSACT , T ) -> H(FPCSLP , 0.30)
F(ARADMAL , T ) & H(FRADM,*:0.0) -> H(FPCSLP , 0.40)
*END
#
# NO.2 FPRPRC,FPRHTR
#
*IF
H(FPRZ , 0.1:1.0)
*THEN
(1 : F(APREPRL,T),F(APREPRH,T)) -> H(FPRPRC, 0.15),H(FPRHTR, 0.10)
(1 : F(ALEVPRL,T),F(ALEVPRH,T)) -> H(FPRPRC, 0.15),H(FPRHTR, 0.10)
(1 : F(ATEPRTH,T),F(APRERTH,T)) -> H(FPRPRC, 0.15)

```

```

F(HTRBUP ,T) & F(HTRPRP ,F) -> H(FPRHTR, 0.60)
*END
#
# NO.3      FCRDCS,FCRDDM
#
*IF
H(FCRD , 0.1:1.0)
*THEN
F(ACRDWIL, T) -> H(FCRDCS , 0.30)
(1 : F(SPRFXH , T),F(ACRDBTM , T)) -> H(FCRDCS , 0.20)
F(SPRFXH , T) -> H(FCRDDM , 0.40)
*END
#
# NO.4      FCVCLC,FCVCCG
#
*IF
H(FCVCS, 0.1:1.0)
*THEN
(2 : F(APREVTH,T),F(APREVTL,T),F(ATEMVTH,T)) -> H(FCVCLC , 0.60)
F(AFLCHAL, T) -> H(FCVCCG , 0.40)
F(ATEMLDH, T) -> H(FCVCCG , 0.40)
*END
#
# NO.5      FSTMPC,FSTMLL
#
*IF
H(FSTM , 0.1:1.0)
*THEN
F(VH104 , T) -> H(FSTMPC , 0.20)
F(AFLSLH , T) -> H(FSTMPC , 0.25),H(FSTMLL , 0.25)
F(ESSACT , T) -> H(FSTMLL , 0.15)
F(APRECON , T) -> H(FSTMLL , 0.20)
F(ACRDSTP , T) -> H(FSTMLL , 0.20)
(1 : F(AINISI ,T),F(AINISG,T)) -> H(FSTMLL , 0.15)
*END
#
# NO.6      TURTRP,GENTRP
#
*IF
H(FTRG , 0.1:1.0)
*THEN
(1 : F(ATRPTR,T),F(STURTR,T)) -> H(TURTRP , 0.40)
F(AOFTDRL,T) -> H(TURTRP , 0.10)
(1 : F(ATRPGE,T),F(AVOLRGT,T)) -> H(GENTRP , 0.40)
F(ATRPEXF,T) -> H(GENTRP , 0.10)
F(ATRPTR , T) AFTER F(ATRPGE , T) -> H(TURTRP , -0.80)
F(ATRPGE , T) AFTER F(ATRPTR , T) -> H(GENTRP , -0.80)
F(ATRPGE , T) AFTER(0.35:0.45) F(ATRPTR , T) -> H(FGENES, -0.3)
*END
#
# NO.7      FCONCW
#
*IF
H(FCOND, 0.1:1.0)
*THEN
F(STURTR , T) -> H(FCONCW , 0.30)
F(APRECVL, T) -> H(FCONCW , 0.50)
F(TURCPH , T) -> H(FCONCW , 0.50)
(1 : F(PMCIRA,F),F(PMCIRB,F)) -> H(FCONCW , 0.60)
*END
#
# NO.8      FSGLCN,FSGFWL

```

```

#
*IF
H(FSGFW, 0.1:1.0)
*THEN
F(SSGLEL, T) -> H(FSGLCN, 0.30)
F(ALESGE, T) -> H(FSGLCN, 0.20)
(1 : F(ALESGBH,T),F(ALESGBL,T)) -> H(FSGLCN, 0.20)
(1 : F(ALESGCH,T),F(ALESGCL,T)) -> H(FSGLCN, 0.20)
F(ALESGCL, T) & F(ALESGBL,T) -> H(FSGFWL, 0.40)
F(SSGLEL, T) -> H(FSGFWL, 0.30)
F(PMAXFW, T) -> H(FSGFWL, 0.10)
*END
#
*IF
H(IHTAX1, 0.4:1.0)
*THEN
(1 : H(FPCSPM, 0.1 : 1.0),H(FPCSLP, 0.1 : 1.0)) -> H(IHTAX2, 0.4)
(1 : H(FPRPRC, 0.1 : 1.0),H(FPRHTR, 0.1 : 1.0)) -> H(IHTAX2, 0.4)
(1 : H(FCRDCS, 0.1 : 1.0),H(FCRDDM, 0.1 : 1.0)) -> H(IHTAX2, 0.4)
(1 : H(FCVCLC, 0.1 : 1.0),H(FCVCCG, 0.1 : 1.0)) -> H(IHTAX2, 0.4)
(1 : H(FSTMPC, 0.1 : 1.0),H(FSTMML, 0.1 : 1.0)) -> H(IHTAX2, 0.4)
(1 : H(FTURCS, 0.1 : 1.0),H(FGENES, 0.1 : 1.0)) -> H(IHTAX2, 0.4)
(1 : H(FCONCW, 0.1 : 1.0),H(FSGLCN, 0.1 : 1.0)) -> H(IHTAX2, 0.4)
H(FSGFWL, 0.1 : 1.0) -> H(IHTAX2, 0.4)
*END
#
# THIRD GRADE OF TAXONOMY
#
**KU
*COND H(IHTAX2, 0.4 : 1.0)
#
# NO.1 FAILURE OF PRIMARY COOLANT SYSTEM
#
**FH RULES
#
#F(VL1460A,T) & F(FLCHAG,0:15) & F(ARADMAL,T) -> H(FRADM, 0.50)
#
*HH RULE
#
*IF
H(FPCSPM, 0.4:1.0)
*THEN
(1 : F(YREVFL1,T),F(YREVFL2,T),F(YREVFL3,T)) &
(1 : F(ALEVPRH,F),F(APREPRH,F)) -> H(RCSPL1,0.70)
(1 : F(YTEMCH1,T),F(YTEMCH2,T),F(YTEMCH3,T)) -> H(RCSPL1, 0.70)
(2 : F(YREVFL1,T),F(YREVFL2,T),F(YREVFL3,T)) &
(2 : F(YTEMCH1,T),F(YTEMCH2,T),F(YTEMCH3,T)) -> H(RCSPL2, 0.90)
(1 : F(YREVFL1,T),F(YREVFL2,T),F(YREVFL3,T)) &
(1 : F(ALEVPRH,T),F(APREPRH,T)) &
(1 : F(YTEMCH1,T),F(YTEMCH2,T),F(YTEMCH3,T)) -> H(RCSPSL, 0.90)
F(YREVFL1,F) & F(YREVFL2,F) & F(YREVFL3,F) & F(AFLRCSL,T)
-> H(RCSPLA, 0.70)
F(YTEMCH1,F) & F(YTEMCH2,F) & F(YTEMCH3,F) & F(AFLRCSL,T)
-> H(RCSPLA, 0.70)
F(AFLRCSL, F) -> H(RCSPL1, -0.80)
F(AFLRCSL, F) -> H(RCSPL2, -0.8)
F(AFLRCSL, F) -> H(RCSPLA, -0.80)
F(AFLRCSL, F) -> H(RCSPSL, -0.8)
*END
#
*IF
H(FPCSLP, 0.4:1.0)

```

```

*THEN
F(APREPRL , T ) & F(ESSACT , T ) -> H(RCSUTR , 0.30 )
F(APREPRL , T ) & F(ESSACT , T ) -> H(RCSLSL , 0.30 )
(1 : F(PRCNT , 0.65:* ) , F(TECONT , 45:* ) ) -> H(RCSLSL , 0.50 )
(1 : F(AFLCHAH , T ) , F(FLCHAG , 20:* ) ) -> H(RCSUTR , 0.30 )
(1 : F(PRCNT , 0:0.8 ) , F(TECONT , 0:43 ) ) & H(FRADM , -1:1 ) & F(ARADMAL , T )
-> H(RCSLSL , -0.7 )
(1 : F(AFLCHAH , T ) , F(FLCHAG , 20:* ) ) -> H(RCSLSL , 0.3 )
F(PMCNSP , T ) -> H(RCSLSL , 0.50 )
(1 : F(ATEPRLH , T ) , F(VP455C , T ) ) & F(PREPRZ , 0:164 ) -> H(RCRVSO , 0.90 )
F(PREPRZ , 0:164 ) -> F(VP455C , F )
F(PREPRZ , 164:* ) -> F(VP455C , T )
F(ATEPRLH , T ) -> F(VP455C , T )
F(ATEPRLH , F ) -> F(VP455C , F )
#H(RCRVSO , 0.8:* ) -> F(VP455C , T )
F(AINISI , T ) -> H(RCRVSO , 0.20 )
H(FRADM , *:0.1 ) & F(ARADMAL , T ) -> H(RCSUTR , 0.70 )
*END
#
# NO.2 FAILURE OF PRESSURIZER SYSTEM
#
*IF
H(FPRPRC , 0.4:1.0 )
*THEN
(1 : F(SPRZPL , T ) , F(APREPRL , T ) ) -> H(PSPVO1 , 0.90 )
F(ATEPRLH , T ) & F(APREPRH , T ) -> H(PSPVC2 , 0.90 )
(1 : F(ATEPRLH , T ) , F(VP455C , T ) ) & F(ALEVPRH , T ) -> H(CPRPFH , 0.90 )
F(ATEMLDH , T ) & F(AFLCHAL , T ) -> H(CPRPFL , 0.70 )
F(VP455C , T ) -> H(CPRPFL , 0.30 )
*END
#
*IF
H(FPRHTR , 0.4:1.0 )
*THEN
(2 : F(LEVPRT , 0:0.05 ) , F(PREPRZ , 155.3:* ) ) -> H(PHTBFN , 0.60 )
F(VP455A , T ) & F(VP455B , T ) -> H(PHTBFN , 0.70 )
F(PREPRZ , 0:155 ) & F(PREPRZ , 165:* ) -> H(PHTPFF , 0.90 )
*END
#
# NO.3 FAILURE OF CONTROL ROD SYSTEM
#
*IF
H(FCRDCS , 0.4:1.0 )
*THEN
F(ACRDBTM , T ) & F(SCRAM , F ) -> H(CRDU CI , 0.60 )
F(SCRAM , F ) & F(LCRODA , -1:5 ) & F(LCRODB , -1:5 ) -> H(CRDU CI , 0.50 )
F(SCRAM , F ) & F(LCRODC , -1:5 ) & F(LCRODD , -1:5 ) -> H(CRDU CI , 0.50 )
F(ACRDWIL , T ) -> H(CRDUCW , 0.20 )
F(ACRDWIL , T ) -> H(CRDMDF , 0.20 )
(1 : F(QFLUX , 10:* ) , F(ACRDBTM , F ) ) & F(SCRAM , T ) -> H(CRDATAF , 0.95 )
F(AATEAVRF , T ) & F(APREPRH , T ) -> H(CLATEH , 0.90 )
*END
#
*IF
H(FCRDDM , 0.4:1.0 )
*THEN
F(ACRDBTM , T ) & F(SCRAM , F ) & F(LCRODA , 0:5 ) -> H(CRDDR1 , 0.90 )
F(AFLCHAH , T ) & F(SPRFXH , T ) -> H(CRDEJ1 , 0.60 )
F(VL1460A , F ) -> H(CRDEJ1 , 0.30 )
F(SPRFXH , T ) AFTER F(ACRDSTP , T ) -> H(CRDWDF , 0.90 )
F(ACRDWIL , T ) -> H(CRDMAS , 0.20 )
F(SPCROD , 50:* ) -> H(CRDMAS , 0.90 )

```

```

F(ACRDWIL , T ) -> H(CRDMIS , 0.20 )
F(SPCROD,6:10 ) -> H(CRDMIS , 0.90 )
*END
#
# NO.4 FAILURE OF CVCS
#
*IF
H(FCVCLC , 0.4:1.0 )
*THEN
(F(HTRBUP,F) & F(HTRPRP ,F)) AFTER F(HTRBUP,T) -> H(CVTLFH , 0.50 )
F(AFLCHAL , T ) -> H(CVTLFH , 0.30 )
F(ALEVPRL , T ) -> H(CVTLFH , 0.30 )
F(VL1460A , T ) -> H(CVTLFH , 0.30 )
F(V1115B , T ) & F(V1115C , F ) -> H(CVTLFL , 0.60 )
F(ACRDWIL , T ) -> H(CVTLFL , 0.40 )
*END
#
*IF
H(FCVCCG , 0.4:1.0 )
*THEN
F(AFLCHAL , T ) -> H(CHAGFL , 0.40 )
F(ATEMLDH , T ) -> H(CHAGFL , 0.30 )
F(FLCHAG,0 : 1) -> H(CHAGFL , 0.80 )
*END
#
# NO.5 FAILURE OF STEAM LINE SYSTEM
#
*IF
H(FSTMPC , 0.4:1.0 )
*THEN
F(APRSLI , T ) AFTER F(ALESGE , T ) -> H(CSHPFH , 0.60 )
F(AVISOWF , F ) -> H(CSHPFH , 0.20 )
F(VH104 , T ) -> H(CSHPFH , 0.10 )
F(ACRDWIL , T ) -> H(CEHCAL , 0.10 )
*END
#
*IF
H(FSTMLL , 0.4:1.0 )
*THEN
F(ARUNBTR , T ) -> H(SDVLA , 0.10 )
F(ACRDSTP , T ) -> H(SDVLA , 0.10 )
F(VL102A , T ) -> H(SDVLA , 0.40 )
F(APRECON , T ) & F(APREPRH , T ) -> H(SGSLL , 0.60 )
F(AVISOLC , T ) & F(APRSLI , T ) -> H(SGSLL , 0.40 )
(1 : F(AINISI ,T),F(AINISG ,T)) -> H(SGSLL , 0.40 )
F(PMCNSP , T ) -> H(SGSLL , 0.30 )
F(PMRCIR , T ) -> H(SGSLL , 0.30 )
F(PMSGFW , F ) -> H(SGSLL , 0.30 )
*END
#
# NO.6 FAILURE OF TURBINE GENERATOR SYSTEM
#
*IF
H(FTURCS , 0.4:1.0 )
*THEN
F(QGENETR,820:845) AFTER F(QGENETR,845:850) AFTER F(QGENETR,835:845)
-> H(TGVFO1, 0.80)
F(QGENETR ,730:770) AFTER F(QGENETR ,820:*) -> H(TGVFC1 , 0.80 )
F(AFLSLH , T ) -> H(TGVFC1 , 0.20 )
F(ACRDSTP , T ) -> H(TURRBF , 0.10 )
F(ATRPGE,T) AFTER F(ATRPTR , T ) -> H(TURTRP , 0.60 )
F(SCRAM ,T) AFTER F(TURTRIP, T ) -> H(TURTRP , 0.60 )

```



```

F(TURTRIP,T) AFTER F(ATRPEXF, T) -> H(TURTRP, 0.30 )
F(VT105A ,F) AFTER F(VT105A , T) -> H(TURTRP , 0.10 )
*END
#
*IF
H(FGENES , 0.4:1.0 )
*THEN
F(ATRPTR,T) AFTER F(ATRPGE,T) -> H(GENTRP , 0.60 )
F(SCRAM ,T) AFTER F(TURTRIP,T) AFTER F(ATRPGE,T) -> H(GENTRP , 0.60 )
*END
#
# NO.7 FAILURE OF CONDENSATION SYSTEM
#
*IF
H(FCONCW , 0.4:1.0 )
*THEN
F(PRCOND, 0.2:0.32 ) -> H(CCWPL , 0.80 )
F(PRCOND, 0.2:0.4) AFTER F(PRCOND, 0.4:*) -> H(CCWTL , 0.80 )
F(APRECVL, T) -> H(CCWTL , 0.70 )
*END
#
# NO.8 FAILURE OF SG FEEDWATER SYSTEM
#
*IF
H(FSGLCN , 0.4:1.0 )
*THEN
F(ALESGCL , T) & F(ALESGBL , F) -> H(CSGLFH , 0.40 )
F(FLMFW3 ,0:5) & F(LEVSG3 ,0:5) -> H(CSGLFH , 0.50 )
F(ALESGCH , T) & F(ALESGBH , F) -> H(CSGLFL , 0.40 )
F(FLMFW3,100:*) & F(LEVSG3 ,90:*) -> H(CSGLFL , 0.60 )
*END
#
*IF
H(FSGFWL , 0.4:1.0 )
*THEN
F(SCRAM , F) & F(PMSGFW , T) -> H(FWFPL , 0.50 )
(1 : F(TURTRIP,T),F(SSGLEL,T)) -> H(FWFTL , 0.10 )
(2 : F(FLMFW1 ,0:5),F(FLMFW2 ,0:5),F(FLMFW3 ,0:5)) &
(2 : F(VF478 ,T),F(VF488 ,T),F(VF498 ,T)) AFTER F(QGENETR ,100:*)
-> H(FWFTL , 0.5)
(2 : F(FLMFW1 ,0:5),F(FLMFW2',0:5),F(FLMFW3 ,0:5)) & F(TEMAVE,288:*)
-> H(FWFTL , 0.5)
F(ALESGBL,T) & F(ALESGCL,T) & F(VF488,T) & F(VF498,T) -> H(FWFTL, 0.80)
F(VL123A , T) -> H(HPHDPL , 0.80 )
*END
#
# TREATMENT
#
**KU
*COND H(IHTAX2,0.4:1.0)
*HH RULES
*IF
F(NPPA , T )
*THEN
H(RCSPL1 , 0.6:1 ) -> H(TREAT01, 0.96 )
H(RCSPL2 , 0.6:1 ) -> H(TREAT02, 0.96 )
H(RCSPLA , 0.6:1 ) -> H(TREAT03, 0.96 )
H(RCSPSL , 0.6:1 ) -> H(TREAT04, 0.96 )
H(RCSUTR , 0.6:1 ) -> H(TREAT05, 0.96 )
H(RCSLSL , 0.6:1 ) -> H(TREAT06, 0.96 )
H(RCRVSO , 0.6:1 ) -> H(TREAT07, 0.96 )
#

```

```
H(PSPV01 , 0.6:1 ) -> H(TREAT08, 0.96 )
H(PSPVC2 , 0.6:1 ) -> H(TREAT09, 0.96 )
H(CPRPFH , 0.6:1 ) -> H(TREAT10, 0.96 )
H(CPRPFL , 0.6:1 ) -> H(TREAT11, 0.96 )
H(PHTBFN , 0.6:1 ) -> H(TREAT12, 0.96 )
H(PHTPFF , 0.6:1 ) -> H(TREAT13, 0.96 )
#
H(CRDUCI , 0.6:1 ) -> H(TREAT14, 0.96 )
H(CRDUCW , 0.6:1 ) -> H(TREAT15, 0.96 )
H(CRDMDF , 0.6:1 ) -> H(TREAT16, 0.96 )
H(CRDATA , 0.6:1 ) -> H(TREAT17, 0.96 )
H(CLATEH , 0.6:1 ) -> H(TREAT18, 0.96 )
H(CRDDR1 , 0.6:1 ) -> H(TREAT19, 0.96 )
H(CRDEJ1 , 0.6:1 ) -> H(TREAT20, 0.96 )
H(CRDWDF , 0.6:1 ) -> H(TREAT21, 0.96 )
H(CRDMAS , 0.6:1 ) -> H(TREAT22, 0.96 )
H(CRDMIS , 0.6:1 ) -> H(TREAT23, 0.96 )
#
H(CVTLFH , 0.6:1 ) -> H(TREAT24, 0.96 )
H(CVTLFL , 0.6:1 ) -> H(TREAT25, 0.96 )
H(CHAGFL , 0.6:1 ) -> H(TREAT26, 0.96 )
#
H(CSHPFH , 0.6:1 ) -> H(TREAT27, 0.96 )
H(CSHPFL , 0.6:1 ) -> H(TREAT28, 0.96 )
H(CEHCAL , 0.6:1 ) -> H(TREAT29, 0.96 )
H(SDVLA , 0.6:1 ) -> H(TREAT30, 0.96 )
H(SGSLI , 0.6:1 ) -> H(TREAT31, 0.96 )
#
H(TGVFO1 , 0.6:1 ) -> H(TREAT32, 0.96 )
H(TGVFC1 , 0.6:1 ) -> H(TREAT33, 0.96 )
H(TURRBF , 0.6:1 ) -> H(TREAT34, 0.96 )
H(TURTRP , 0.6:1 ) -> H(TREAT35, 0.96 )
H(GENTRP , 0.6:1 ) -> H(TREAT36, 0.96 )
#
H(CCWPL , 0.6:1 ) -> H(TREAT37, 0.96 )
H(CCWTL , 0.6:1 ) -> H(TREAT38, 0.96 )
#
H(CSGLFH , 0.6:1 ) -> H(TREAT39, 0.96 )
H(CSGLFL , 0.6:1 ) -> H(TREAT40, 0.96 )
H(FWFPL , 0.6:1 ) -> H(TREAT41, 0.96 )
H(FWFTL , 0.6:1 ) -> H(TREAT42, 0.96 )
H(HPHDPL , 0.6:1 ) -> H(TREAT43, 0.96 )
*END
#
```