ICOT Technical Memorandum: TM-0927

TM-0927

知識検証システム
KNOV操作説明書

田中立二，戸辺啓子，山尾雅利(東芝)

July, 1990

©1990, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

知識検証システム　ＫＮＯＶ

操作説明書

田中立二　戸辺啓子　山尾雅利

株式会社東芝

１９９０年４月

目次

[1] 概要

知識プログラミング・ソフトウェアでの知識ベース構築においては、専門家からの知識の抽出・知識の確認・整理・構造化を支援する知識獲得機能、および獲得した大量の知識間の整合性を検証する機能が必須である。これらを総合的に支援する知識ベース構築支援システムの一環として、診断・解析問題を対象とした知識ベースの整合性を保証するための知識検証システムを開発した。

知識検証の問題は、不完全かつ矛盾を含む知識ベースを、完全で整合性のある知識ベースに修正、整理して行き、問題解決を正しく行える様にする問題である。知識ベースに含まれる矛盾ルール、不完全ルールとそれを解消するためのルール修正案を仮説と考え、仮説推論を用いて、矛盾解消戦略に基づく仮説生成とその検証を行うことにより、知識ベースを洗練化し整合性のあるものにしていく動的検証方式による知識検証システムＫＮＯＶ(KNOwledge Verification system)を開発した。また、その知識ベースの矛盾検出・修正を支援する機能については、仮説集合に基づいた知識の一貫性維持を行うＡＴＭＳを利用した知識管理機構として実現した。

[2] 知識検証方式

2.1 知識獲得支援システム

プラント制御システムの故障診断などの診断・解析問題を対象とした知識獲得支援システム構成を図2-1に示す。知識獲得支援システムは、知識獲得部と知識検証部から構成されそれぞれ以下に示す特徴を持つ。

知識獲得システム
- ・対象問題の知識構造を抽出した知識フレームによる知識獲得支援
- ・知識概念および階層間の整合性に基づく知識洗練化戦略

知識検証システム
- ・動的検証方式に基づいた矛盾知識検出と仮説生成による知識検証
- ・ＡＴＭＳによる知識整合性管理・矛盾知識管理

知識獲得および知識検証で用いられる知識フレームを図2-2に示す。知識フレームは対象領域の診断知識の構造を抽出した知識枠組みであり、以下に示す３階層で構成される。

ルールフレーム
診断ルールを分類・形式化したものであり、対象領域の問題解決の枠組みと、知識記述形式の枠組みの規定

基本概念
診断ルールを記述するための基本要素（語彙）であり、診断対象プラントの運用状態、状態変化、仮説（原因）などの集合。

データフレーム
診断対象の設備機器構成を定義する枠組み。構成要素である機器の属性定義、構成要素間の結合関係とその制約定義形式。

2.2 知識検証

知識検証の目的は、不完全な、矛盾を含む知識ベースを、完全で整合性のある知識ベースに修正、整理して行き、問題解決を正しく行える様にすることである。知識検証の過程は、図2-3の様に示される。

(1)知識誤りの検出と局所化
知識が誤りである事の定義と、その原因の分析（矛盾、不完全、欠如など）、および知識の中での誤りの場所（該当ルール、ルール内の記述）を求める

(2)仮説生成
知識の誤りを修正するための代替ルール案の作成。

(3)仮説検証
代替ルール案の正しい事の確認。推論を実行し他のルールとの干渉（矛盾）、抽象度の適切さの確認を行い正しい結果の得られる事を示す。

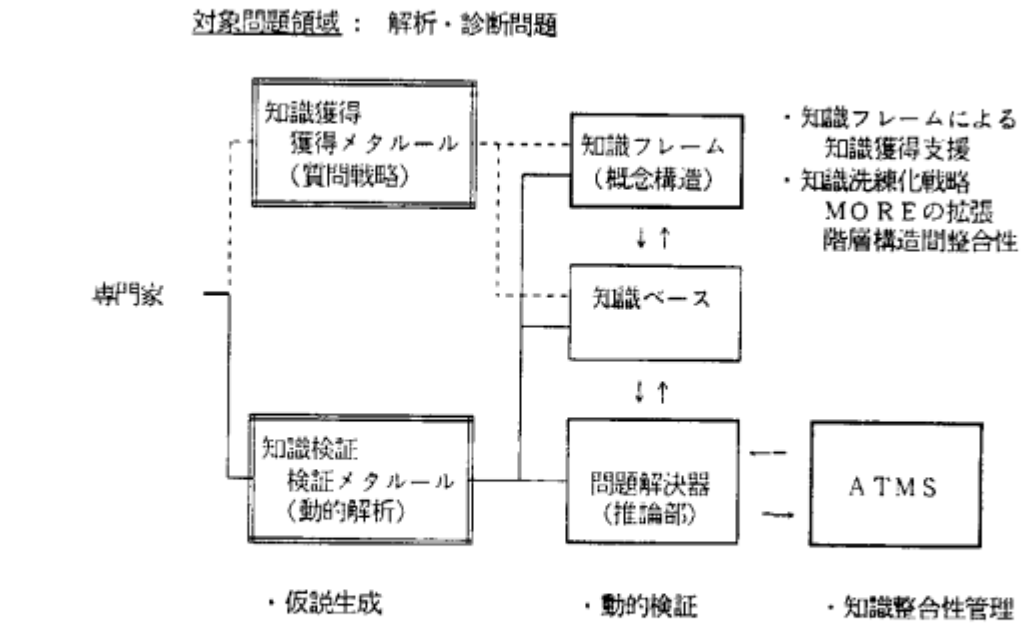


図2-1 知識獲得支援システム

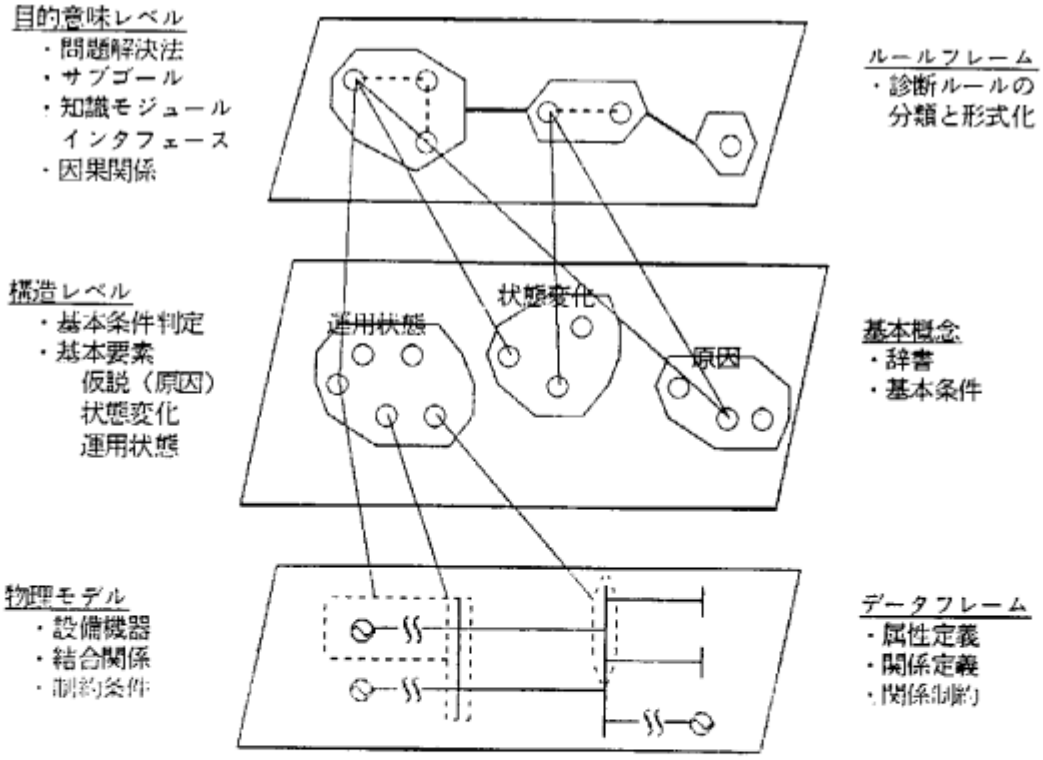


図2-2 知識フレーム

これらの各過程で、どのように行なえば完全に検証が出来るか、またどのような手順で行なえ　ば効率良く検証が出来るか、についての知識検証戦略が必要とされる。これは具体的には、テストケースの集合と、テスト手順として与えられるが、一般には上記各過程の途中で状況に応じて決めていく場合が多い。

これに対する、知識検証の機能についての処理の流れと、それに必要な情報の関係を示すと図2-4の様に考える事ができる。これからわかる様に、知識ベースに含まれる矛盾ルール、不完全ルールとそれを解消するためのルール修正案を仮説と考え、仮説推論を用いて矛盾解消戦略に基づく仮説生成とその検証を行なう事により、知識ベースを洗練化し整合性のあるものにして行く事が可能となる。知識整合性の管理としては、仮説集合に基づいた知識の一貫性維持を行なうＡＴＭＳを利用する事ができる。

2.3　動的検証方式

検証の対象となる知識に対して、実際に推論を実行する事で知識検証を行う。すなわち推論の過程・結果の情報を用いて、矛盾検出、仮説（ルール修正案）生成、仮説検証を行う。

知識の誤り（矛盾）の検出と局所化については、その対象とする知識階層に応じて行なう。診断問題の知識ベースの場合には、診断ルール、ルールを構成する基本概念、の順に調べて行く。知識誤りの検出は、これらの各階層の知識単位を推論し、その結果生じる矛盾を検出する事で行なう。これは各推論の実行における入力と出力を調べる事により行なう。

診断ルール（あるいは基本概念）のどの部分が悪いかを検出するためには、個々のルールの機能（入力データ、出力データ、結果としての真偽）を仮説と考え、ルールを実行させた時に本来出力すべき結果を観測事象と考えれば、ＡＴＭＳを用いて検出・同定する事ができることがわかっている。

仮説生成と仮説検証は、ルールをどの様に修正したらどの様な推論を行うか、また推論の結果あるいは中間結果は正しいかの判断を行う。ここで考える仮説とは、あるルールの集合を想定すれば正しい診断結果が得られるであろうというものである。誤りのある部分に関してルールの修正案を作成して行き、矛盾が無くなり正しい結果の出されるまで修正を繰り返す。

仮説生成についても、知識の階層構造から、ルールと基本概念の２階層の仮説生成と考える事ができる。また仮説生成については、診断結果と観測事実が異なる場合に、観測事実に合うように、ルールの結論部・条件部を組み替えて行き、ルール修正案を作成する。

仮説生成についての制約条件としては以下に示す様に、知識フレームと、対象領域の深い知識を反映した仮説生成知識を与えるものとする。これらは問題領域に依存している。

・知識修正（仮説生成）のための戦略知識
・知識フレーム　　..知識構造の制約を利用したルールパターンであり、
　　ルールを構成する基本概念の集合を決める。
・仮説生成知識　　..一般化、特殊化、基本概念どうしの結合関係などの制約知識

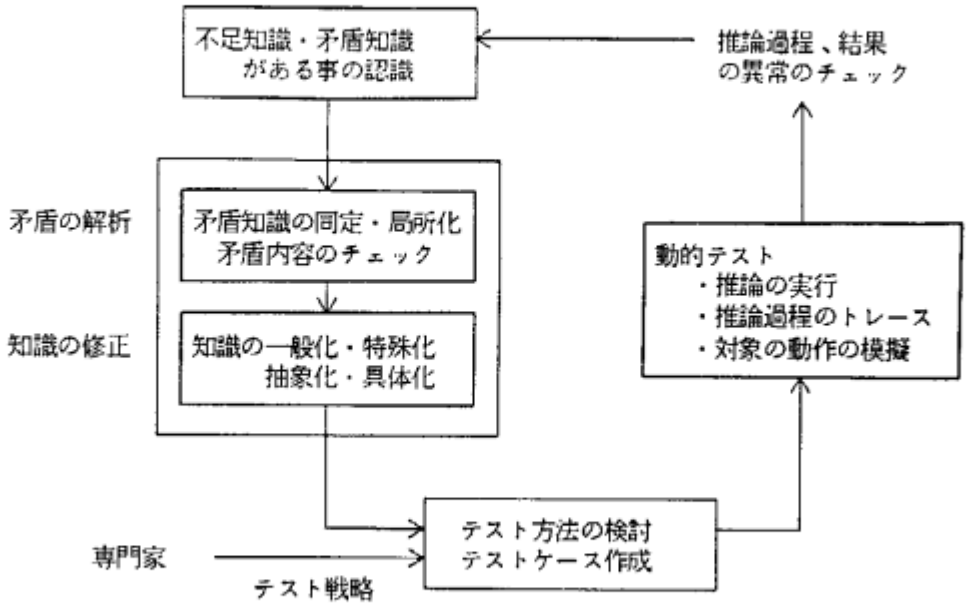


図2-3　知識検証の手順

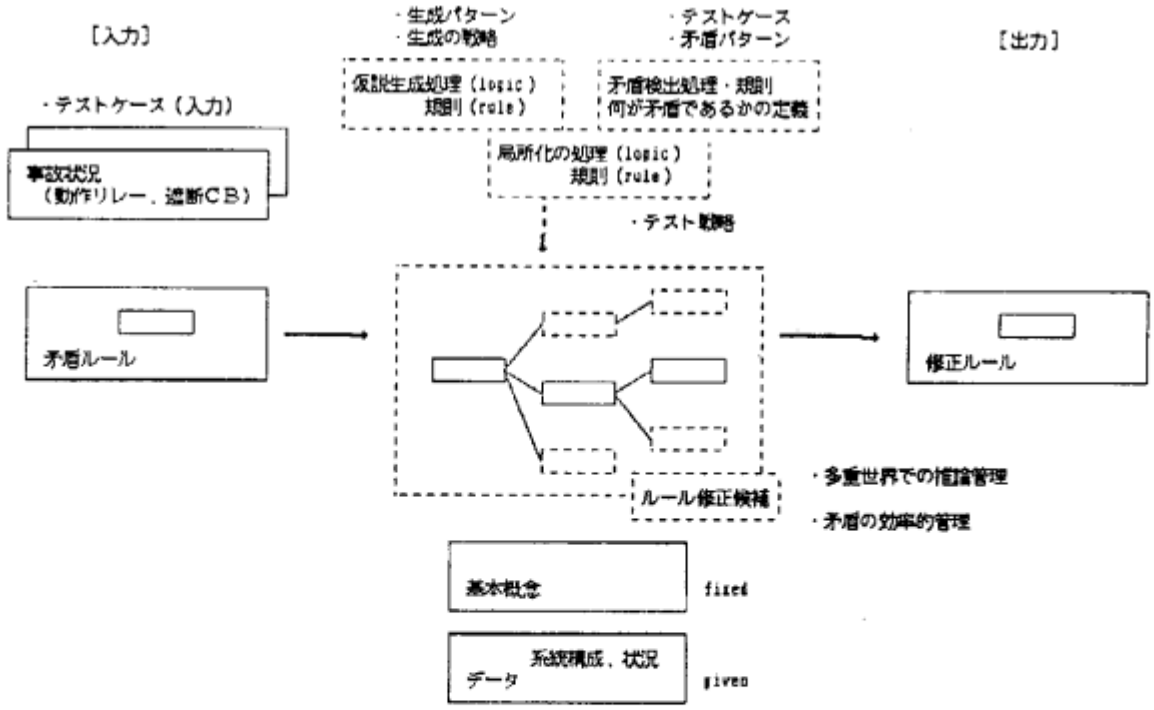


図2-4　知識検証の処理の流れ

仮説検証については生成された仮説（ルール）を用いて診断の推論を行い、矛盾が無いかを調べる。これについては既知の正しい入出力パターンと推論結果の比較照合を行なえば良い。結果が一致すれば仮説集合は正しい解の一つであり、一致しない場合には矛盾が検出された事になる。矛盾が検出された場合には図2-3に示した様に仮説生成、仮説検証のサイクルを繰り返す事となる。

2.4 知識管理方式

生成される仮説の数とその検証（推論の実行回数）は指数的に増大する可能性があるが、これをＡＴＭＳを利用することにより効率的に管理する事ができる。これは、ＡＴＭＳを用いて矛盾知識を管理し、仮説生成においてはすでに矛盾すると分かっている仮説は作成しない様にし、仮説検証時には過去に失敗した推論は行わない様にするためである。

知識検証における仮説推論の場合には、次の様に考える事で知識修正の支援としてＡＴＭＳを適用できる。

- 知識（ルール）修正問題を、基本概念を構成要素とした組合せ問題と考える
- ルール修正のパターンを一つのコンテキストとして考える
- 各コンテキストの矛盾はルール集合を実行する事で検出する
- 矛盾の無いコンテキストが求める知識の修正案となる

[3] 機能仕様

3.1 機能構成

知識検証システムの機能構成は図3-1に示す通りである。この方式は第2章で述べた知識検証方式を実現した形となっている。

知識検証システムは問題解決機構と知識管理機構により構成される。問題解決機構は、知識ベースの矛盾検出機能と矛盾知識の修正案（仮説）生成機能及びその仮説検証機能を持つ。また、矛盾検出、仮説検証を行うための推論機能を持つ。知識管理機構は、ＡＴＭＳを利用して、知識修正案（仮説）の集合と矛盾知識の管理を統一して行う。また、知識検証のための知識として、対象システムの診断知識以外に、診断知識の矛盾を検出し、それを修正するための、「矛盾定義（テストケース）」、「矛盾検出」、「仮説生成」などの知識を持つ。

最初にテストケースと、検証の対象となる診断知識が与えられる。テストケースは事故の状況とそれに対する正しい診断結果を定義したものの集合である。診断知識としては、ルール、基本概念、データの３階層がある。処理は以下に示す様に行われる。

(1)診断用エキスパートシステムの推論機構を用いて検証対象知識を推論する。
推論結果が目的と合致するかどうかをテストケース及び矛盾検出知識により確かめ、もし、合わない場合は、どの知識が原因かを見きわめる。

(2)検討結果に基づいて解決案を作成する。満足できる結果を得るために、仮説生成知識によりルールを修正し、新たな診断用ルールを作成する。この時知識管理機構の矛盾知識情報を用いて不要なルールを作成しない様にする。

(3)新たな診断用ルールの検証を行う。この場合にも知識管理機構の矛盾知識情報を用いて矛盾を導出する様な推論は行わない様に制御する。

(4)仮説生成・検証を繰り返し、元の矛盾ルールを詳細化・具体化あるは抽象化・統合化して最終的に、全てのテストケースに矛盾しない修正ルールを作り出す。

3.2 問題解決機構

問題解決機構の機能構成を図3-2に示す。問題解決機構は、診断推論、矛盾検出、仮説生成の３つの機能から構成される。

(1)診断推論

診断推論では、診断知識、生成仮説（ルール修正案）、テストケースを読み込んで推論を行う。この時、推論情報を知識管理機構に出力しながら推論を行い、同時に推論の過程で既知の矛盾知識をチェックし、それに該当すれば推論を省略する。さらに、推論が終了したら推論結果を出力する。

診断知識に対する推論方式は、次の３種類が用意されている。

・全探査方式

対象となる知識群すべてについて推論を行なう。即ち実行可能なルールを全て実

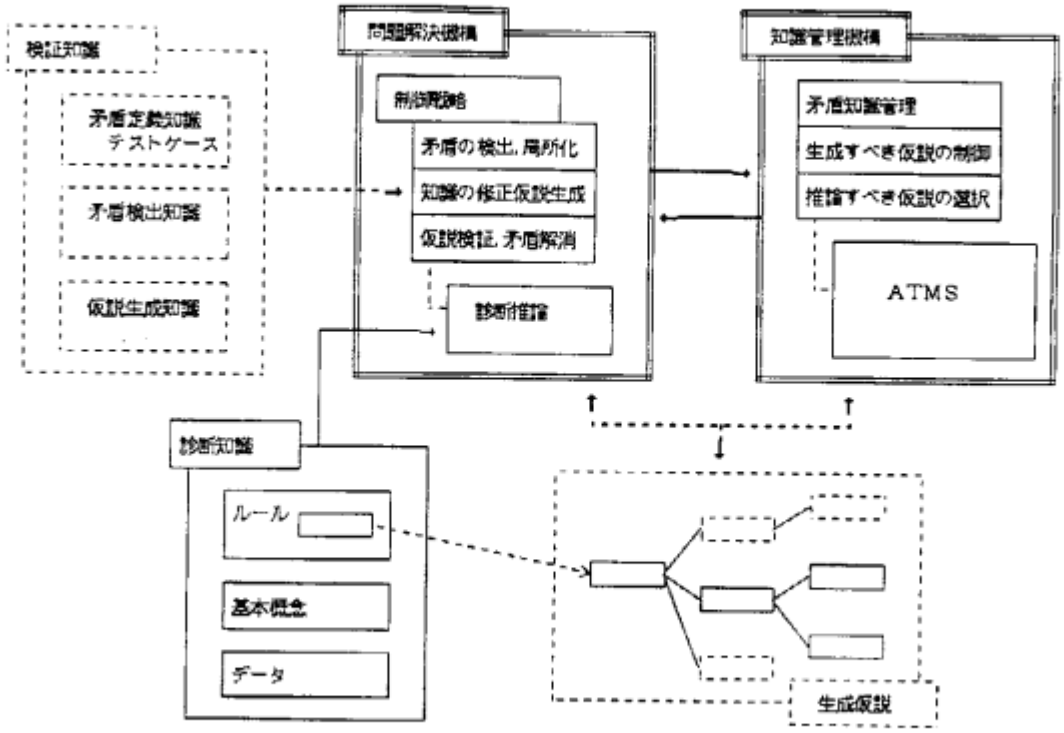


図3-1　知識検証システムの機能構成

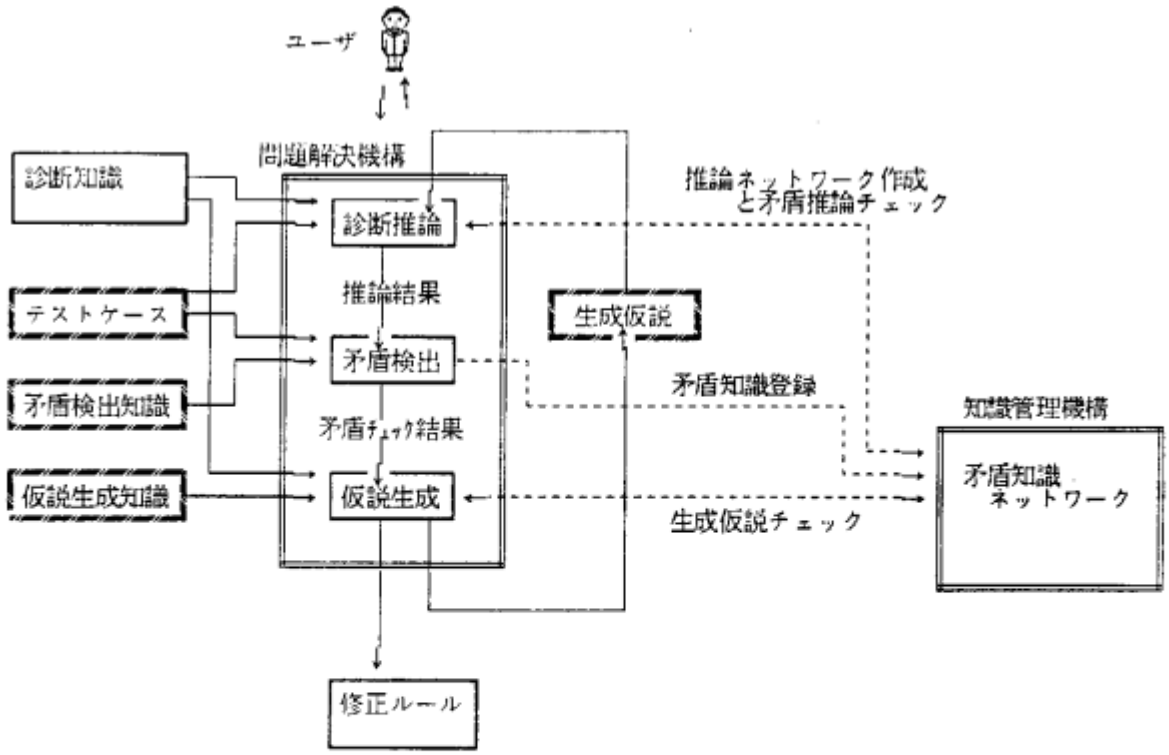


図3-2　問題解決機構の機能構成

行する。

可能性のある故障原因を全て調べ挙げる場合などに用いる。

・優先度探査方式

対象となる知識群の個々のルールに付加されている優先度に注目して、その優先度の高い方から推論を行なう。成立する最初のルールを実行する。

推論は、推論起動時に与えられた限界優先度まで行われる。

・順次探査方式

対象となる知識群の個々のルールについて、知識ベース上の並び順で推論を行なう。条件の成立する最初のルールが推論される。

推論方式は、知識の種類と必要とする処理に応じて選択できる。推論は、テストケース毎に（事故現象とその原因、事故状況）、各々推論を行い、個々の推論情報を管理する。この様子を示したのが図3-3である。

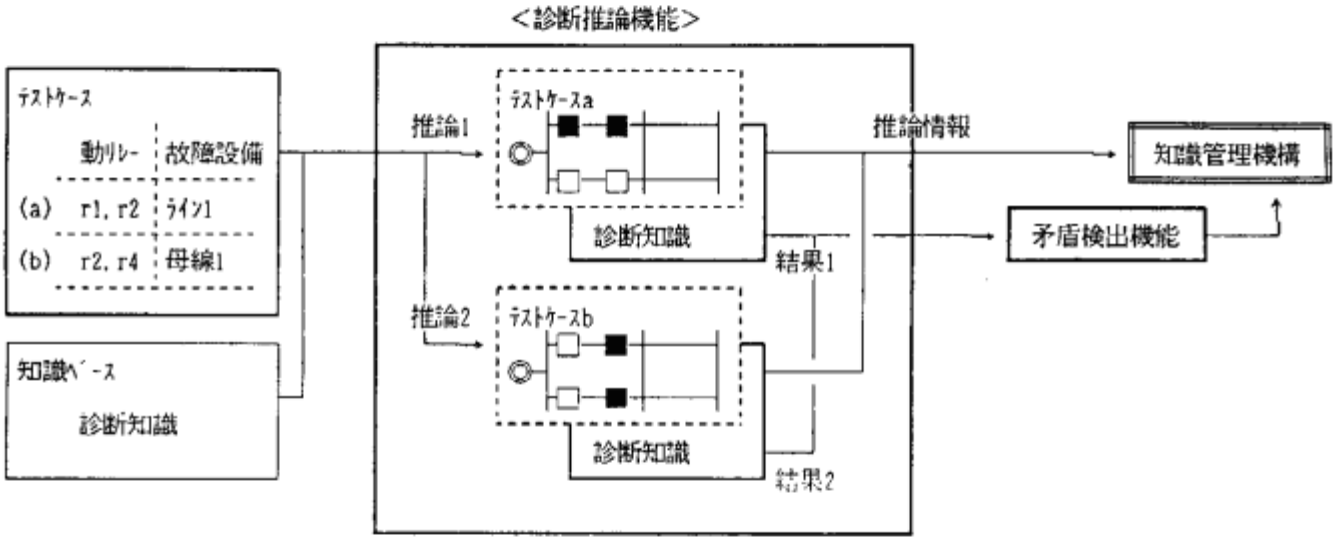


図3-3　推論機能

(2)矛盾検出

矛盾検出では、診断推論機能での推論結果と矛盾検出知識とを使って、誤りの検出を行う。ここで言う「誤り」とは、診断の目的に合った回答（推論結果）が得られないということである。矛盾検出は、推論結果の情報に対して矛盾検出知識が適用できるかどうかにより行われるが、このときのチェック情報はすべて知識管理機構に渡され、矛盾知識として登録され、仮説生成、検証推論時に利用される。

矛盾検出では、診断推論の結果について誤りがないかどうかをチェックする。誤りとしては、テストケース（原因と結果、及び、テスト時の状況）に示された結果と明らかに異なる場合と、問題領域に関する専門知識から結果が矛盾していると推定できる場合がある。後者を矛盾検出知識として定義する。その知識の表現形式は診断知識と同じものを用いる。

矛盾チェックでは　矛盾検出知識を読み込み、推論結果と照らし合わせながら矛盾が検出されるか否かのチェックを行なう。矛盾チェックは、(1)に述べた推論機能を利用して行なう。即ち矛盾検出知識が採用になった場合は対象となる仮説（診断ルール）は矛盾とされる。矛盾検出知識に対する推論手法としては、"全探査推論方式"を用いる。矛盾仮説は知識管理機構によりnogoodデータベースに登録される。矛盾チェックの概要を、図3-4に示す。同図において、①まず、あるルールの採用結果についての情報を推論機能より受け取る。②次に、そのルール採用についての矛盾がないかどうかを調べる。③そして、矛盾検出知識採用結果をまとめて、診断ルールについての矛盾チェックが終了する。以上の①～③を、推論結果としてのルール採用結果について繰り返す事により矛盾チェックを行なう。

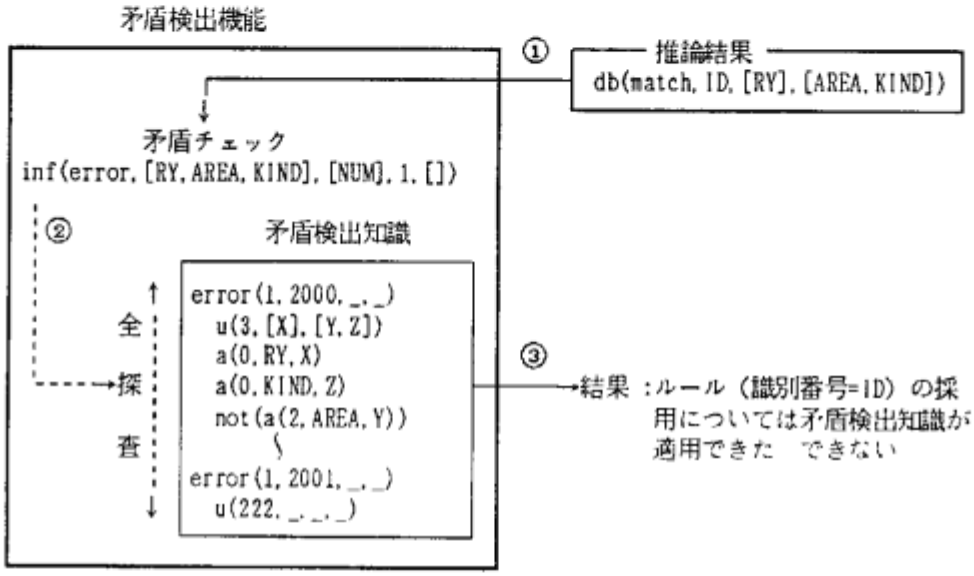


図3-4　矛盾検出機能

(3)仮説生成

仮説生成では、診断知識で用いられている基本概念と仮説生成知識をベースに、矛盾ルールについて、その修正案を作成し、仮説として出力する。さらに、それらの仮説の中から、知識管理機構を用いて無矛盾な仮説を選別し、修正ルールを作成する。仮説生成の処理は以下に示す通りに行われる。

①仮説生成方針決定

矛盾検出により、仮説生成機能が起動される。

まず最初に、以下の項目を明確にして仮説生成の方法を決定する。

・どの仮説生成知識を使うか

・仮説生成知識について、どの方向を採用して仮説を生成するか

より詳しい方向（具体化）／並列なレベルの概念の方向（代替え概念の選択）／上位概念への方向（抽象化）

②仮説生成

仮説生成とは、検証のためのルールの修正案を作成することである。本システムでは、ルールは基本概念を組み合わせることにより表現されている。ルールの修正案の作成は、ルールを組み立てている基本概念を変更することにより行う。基本概念変更の手がかりとして引用されるのが仮説生成知識である。対象とする問題領域についての概念知識の体系を仮説生成知識として定義しておき、それ検索することにより、検証目的として意図するルールの意味の変更、一般化、詳細化等をはかれる項目を見いだし、その項目に対応する基本概念を用いることによりルールの修正を行う。

この様子を示したのが図3-5である。この図の例は、元のルールの2つめの基本概念s0が誤りであった場合に、s0を詳細化した基本概念s1,s2に置き換えられた2つのルール修正案（仮説）が生成される過程を示している。即ち基本概念s0に対応する仮説生成知識のlineを調べ、それを詳細化した概念receive,supplyに展開し、対応する基本概念s1,s2にそれぞれ置き換えたルールを生成する。

③ルール設定

②で仮説を作成したところで知識管理機構を呼び出し仮説の矛盾チェック行う。その結果に基づき生成仮説を選別し修正候補ルールとする。

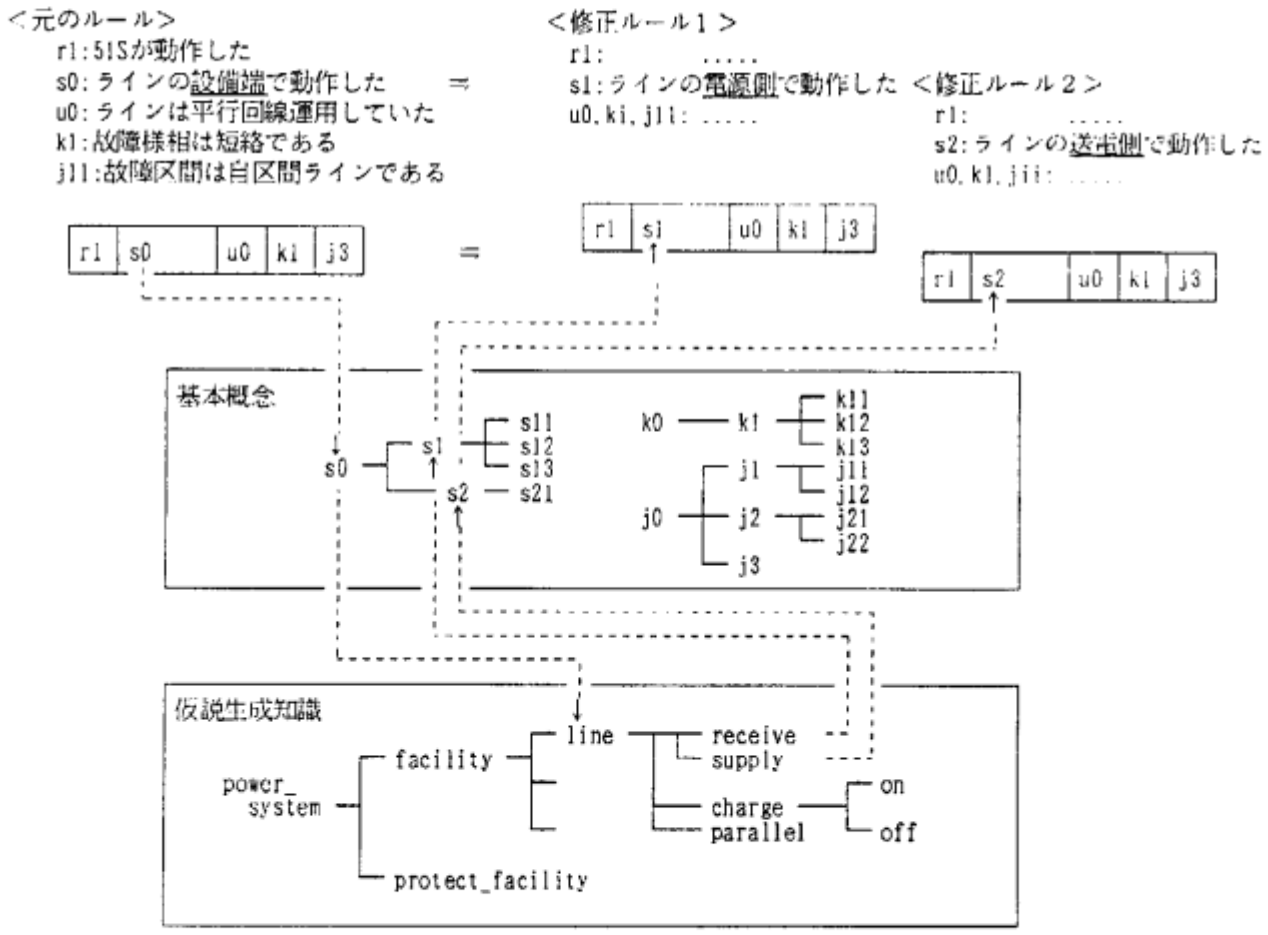
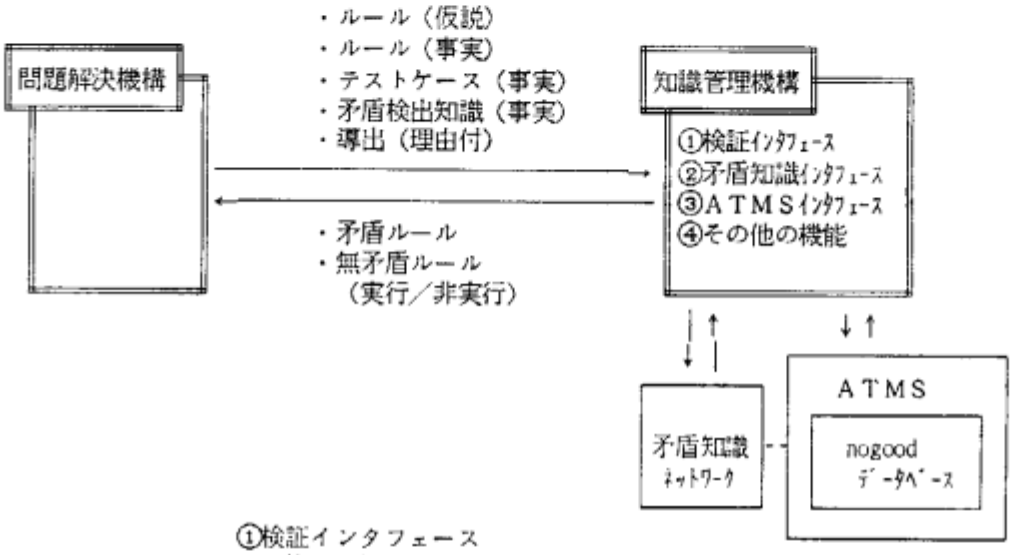


図3-5　仮説生成機能

3.3　知識管理機構

本システムで扱う診断知識は、ルール、基本概念、データの階層関係で定義される。一つの知識モジュール中のルール群は、各項を要素とするAND/OR回路で置き換える事ができる。従ってルールの中の不完全な要素の検出は、多重事故診断の一種と見なされ、故障要素の検出と保存とにＡＴＭＳの手法が有効になる。ただし、知識検証の場合には、観測値とモデルとの不一致からモデルに合わない部品を見いだすのでなく、モデル定義の不具合を検出する事になる。

知識管理機構は、ＡＴＭＳを利用して矛盾を起こすルールと矛盾の理由を管理し、問題解決機構での仮説生成（ルール修正）を支援する。その構成と機能の概要を図3-6に示す。



①検証インタフェース
　推論・検証機構より与えられるルール，データ，推論履歴を変換して矛盾知識ネットワークに登録しＡＴＭＳインタフェースを呼ぶ。
②矛盾知識インタフェース
　仮説生成機構から与えられるルール群とnogoodデータベース内の矛盾ルールと比較し、成否を返す。
③ＡＴＭＳインタフェース
　矛盾知識ネットワークの更新された情報を参照して、ＡＴＭＳのノードの生成・更新・削除を行う。
④その他の機能
　矛盾知識ネットワーク，nogoodデータベースを保存・復元する。

図3-6　知識管理機構の構成

知識管理機構における、ＡＴＭＳのデータ構造と、ＡＴＭＳの機能を利用した矛盾知識ネットワークを図3-7に示す。知識管理機構は推論時と、仮説生成・検証時に問題解決機構から呼び出される。推論時には図3-7に示す様に推論した知識のネットワークが作成されて行く。仮説生成・検証時には、矛盾検出知識により推論結果をチェックし、矛盾があれば該当する仮説集合をＡＴＭＳのnogoodデータベースに登録する。仮説生成時には、矛盾知識として登録されているかのチェックを行い、登録されていれば、問題解決機構は仮説として生成しない様にする。

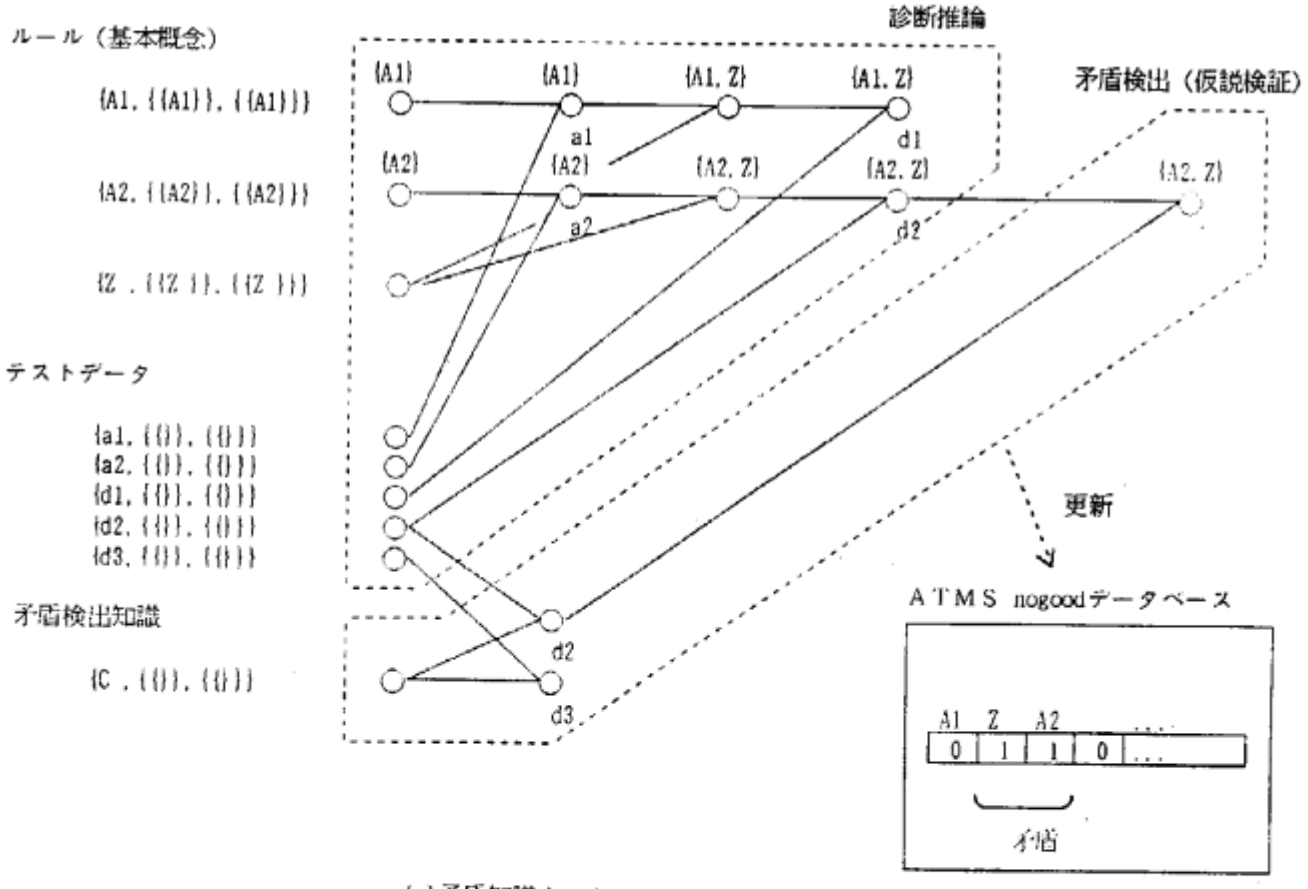
1. ノード
   a. 前提..テストデータ、矛盾検出知識、仮説でない診断ルール
   b. 仮説..診断ルール（基本概念列）
   c. 仮説からの導出
   d. その他の導出
2. 理由付け ：{導出ノード群} または、導出ノード
3. 環境　　：　仮説の組み合わせよりなる集合
4. コンテキスト ：無矛盾な環境の仮説とそれらの仮説から導出されたノードの集合
5. ラベル　：　環境からなる集合で、各ノードが持ち、そのノードが従属する仮説を示す

(a) ＡＴＭＳデータ構造

ルール１（ｘ，ｚ）：ｉｆ Ａ１（ｘ） ｔｈｅｎ Ｚ（ｚ）．
ルール２（ｘ，ｚ）：ｉｆ Ａ２（ｘ） ｔｈｅｎ Ｚ（ｚ）．
矛盾検出知識（ｘ）：ｉｆ　Ｃ（ｘ） ｔｈｅｎ 矛盾．

Ａ１，Ａ２，Ｚはルールを構成する基本概念で仮説となる
ｘはルールの入力変数，ｚは出力変数

(b) 診断ルールと矛盾検出知識の例



(c) 矛盾知識ネットワーク

図3-7　知識検証におけるＡＴＭＳの適用

(1)矛盾知識管理

矛盾知識ネットワークは診断知識の階層構造に従って、ルール単位と基本概念単位の2レベルで作成・管理する。複数のルールが順次推論された場合、ルール間の実行順序（導出）関係と発生した矛盾を、ルールをノードとする矛盾知識ネットワークとして管理する。個々のルールを構成する基本概念については、それぞれの推論時の変数の入出力関係に基づいて基本概念をノードとする矛盾知識ネットワークにより管理する。ルール単位の矛盾知識管理の例を以下に示すが、基本概念の場合も同様である。本例における仮説および事実を以下に示す。

仮説：診断ルール　R(動作リレー[SW,51s]→短絡事故,事故区間は下流3区間[X])
事実：診断ルール　D(動作開閉器[SW,...]→充電区間[Y])
テストケース　T(事故状況　[[s5,50s],[s6,51s]],
動作リレー[s6,51s]→短絡事故,事故区間[BC線1L])
矛盾検出知識　C(事故区間[X]∩充電区間[Y]≠φ→！)

これらの知識の組合せを推論した結果から生成される中間データや矛盾検出の情報は、知識管理機構に渡され、以下の処理が行われる。

1.推論時
・ルールRとテストケースTの入力部分（リレー:51s）がマッチするので、ルールRより事故区間[BC線1L,BC線2L,C母線,CD線1L,CD線2L]が生成される。

2.検証時（矛盾検出）
・ルールDとテストケースTの事故状況（開閉器:s5,s6）がマッチし、ルールDより充電区間[BC線2L,C母線,CD線1L,CD線2L]が生成される。
・この充電区間と1.で生成された事故区間の共通部分として
[BC線2L,C母線,CD線1L,CD線2L]が存在し、矛盾検出知識Cにより矛盾が導出される。

これらの情報を受けた知識管理機構は、ルールおよびデータとＡＴＭＳノードとの矛盾知識ネットワークを作成し、ＡＴＭＳノードを更新する。ＡＴＭＳは、図3-7に示す様に順次ノードを更新し、矛盾が発生すると矛盾の要因を取り除いて、nogoodデータベースに登録する。また、登録した矛盾を発生させた仮説ルールに関する情報を同時に保存する。

(2)生成仮説の矛盾チェック

仮説生成時にnogoodデータベースを参照することで、実際に推論を実行させて検証する前に、矛盾すると分かっている基本概念列の組を含む仮説を排除する。

この時の知識管理機構の処理手順は以下に示す通りである。

1.問題解決機構より仮説を受け取る。
2.仮説のステータスをgoodに設定する。
3.矛盾知識ネットワーク参照のキーとなる述語名、定数を抽出する。
4.矛盾知識ネットワークを調べる。仮説の基本概念列を包含する矛盾ルールがあれば、

その仮説のステータスをnogoodとして探索を終え、次の仮説に移る。また、矛盾知識ネットワークの最後まで仮説の基本概念列を包含する矛盾ルールがない場合も、次の仮説に移る。

5. すべての仮説について探索が終れば、ステータス付きの仮説群を問題解決機構に返す。

3.4 試作の範囲

知識検証システムをＥＳＰ言語により、ＰＳＩ-Ⅱ上に試作した。試作は知識検証の基本方式の有効性を確かめることが目的であるため、以下の制約をもうけた。

1. 知識ベースの誤りは診断ルールにのみ有り、基本概念、データには矛盾が無いものとする。
2. 知識ベースの誤りは同時に一つしか発生しないものとする。
3. 基本概念の矛盾知識ネットワークは、推論実行時でなく、知識入力・生成時に行う。
4. 仮説生成については、仮説生成知識に基づいた、ルールを構成する個々の基本概念の具体化、抽象化を行う。

[4] 操作方法

4.1 システム構成

4.1.1 システム構成

知識検証システムの構成を図4-1に示す。

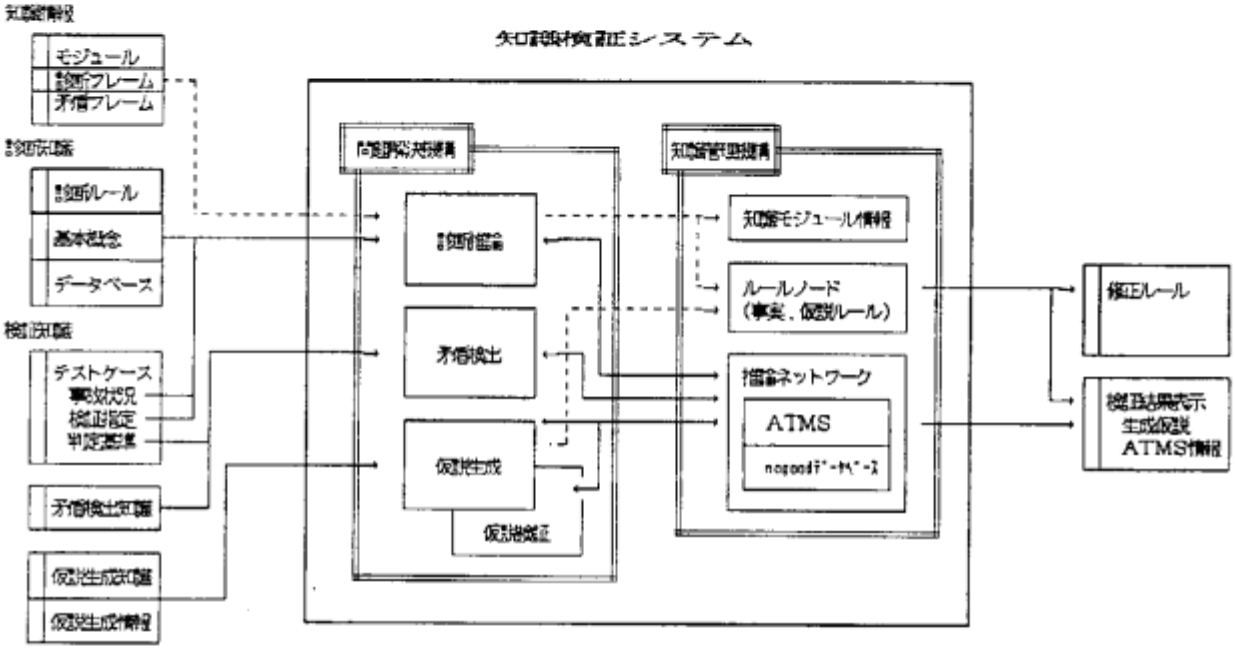


図4.1 知識検証システムの構成

・検証機能

検証は、テストケースにより検証対象ルールの入力値を与える事により開始される。はじめに診断ルールの推論が行われ、推論したルールの出力値がＡＴＭＳに保存される。次に、その出力値を入力とする矛盾検出ルールの推論を行い、テストケースの判断基準により矛盾の有無を調べる。矛盾があれば、すでにＡＴＭＳに保存されている推論データを矛盾とし、矛盾ルールに対して仮説生成知識を用いて仮説生成を行い、生成された仮説に対して検証推論を実行する（仮説検証）。生成仮説のなかでテストケースの判断基準を満足したものが、求める修正ルールとなる。

・修正ルール保存機能

生成された修正ルールを保存する。

・検証結果表示機能

生成された仮説、及びその状態を表示する。

検証による推論結果を、ＡＴＭＳ情報（矛盾／無矛盾）と共に、表示する。

知識検証システムで必要とされるファイルは以下に示す通りである。

知識情報

1. 知識モジュール情報

知識モジュール単位での、入力変数と出力変数を定義する。

2. 診断ルールフレーム

診断ルールで使用する入出力変数、基本概念グループを記述する。

3. 矛盾検出ルールフレーム

矛盾検出ルールで使用する入出力変数、基本概念グループを記述する。

診断知識

4. 診断ルール

診断内容を記述する。これが検証の対象となる。

診断ルールは、拡張子".rule"をつけたファイルに記述する。

5. 基本概念（プリミティブ）

ルールのボディー部を構成する。各基本概念は、基本概念クラスとしてＥＳＰの記述形式でクラスメソッドとして記述する。

6. データベース

診断対象の設備・機器構成を定義する。基本概念で参照する。

データベースは、述語形式で記述する。

検証知識

7. テストケース

診断結果の判断基準、事故状況の設定、検証実行単位の指定を行う。

8. 矛盾検出ルール

矛盾検出を行うためのルールを記述する。拡張子は".crule"とする。

9. 仮説生成知識

診断対象に関する知識を階層で表現したもの。これを用いて仮説生成を行う。

10. 仮説生成情報

仮説生成知識と基本概念の対応づけ、及び仮説生成の方法についての情報を記述する。

4.1.2　プログラム構成

知識検証システムのプログラム構成は、図4-2に示す通りに制御機構、問題解決機構、及び知識管理機構の３つの機構で構成される。

制御機構は、会話、ルール入出力及びデータベースを管理すると共に、問題解決機構、及び知識管理機構を生成し、制御する。

問題解決機構は、推論制御・仮説生成の２つから構成される。推論制御には、推論実行と矛盾検出がある。仮説生成には、仮説生成情報と仮説生成知識の管理およびそれらを用いた仮説生成の３つがある。

知識管理機構は、知識モジュール管理・仮説ルール管理・推論ネットワーク管理の３つで構成される。

・知識モジュール管理

知識モジュールの実行制御および知識モジュール間の変数の受け渡しの管理

・仮説ルール管理

診断知識および生成された仮説をルールと基本概念（プリミティブ）の２レベルの管理

・推論ネットワーク管理

個々の入力データに対する推論過程をネットワークとして記憶し、推論の結果および矛盾の有無をＡＴＭＳを用いて管理する。ＡＴＭＳとしてはＩＣＯＴで開発されたAPRICOT/0を用いている。

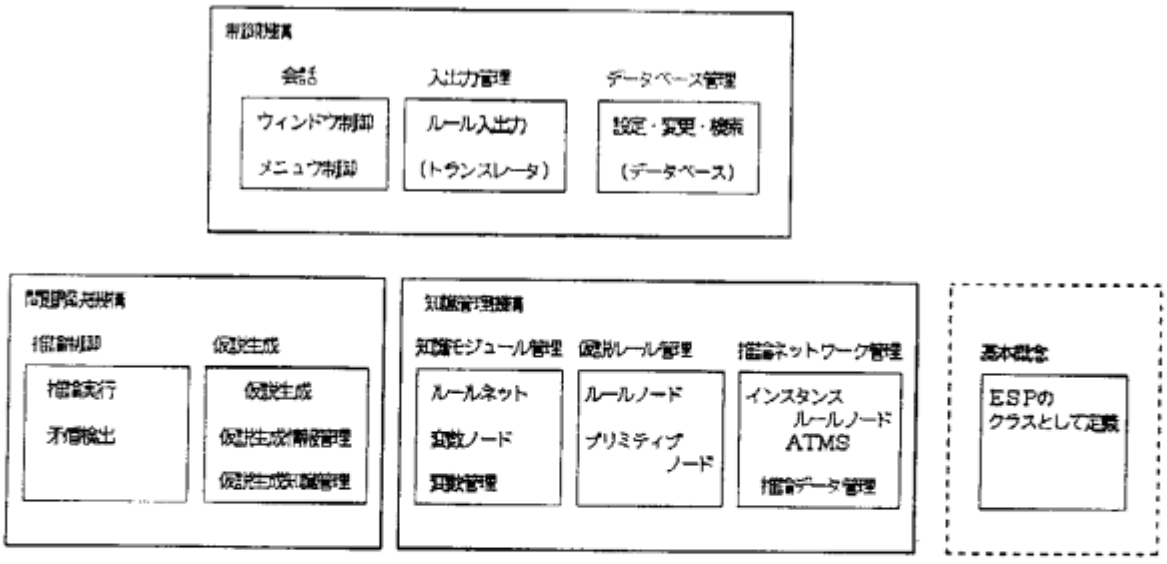


図4-2　知識検証システム　プログラム構成

4.1.3 インストール

以下に示すファイルをカタログし、生成された全クラス、オブジェクトを保存する。

使用するユーザディレクトリ直下のlogin.comファイルのメニュー項目へクラス名=knovを追加すれば、システムメニューより起動できる様になる。

「 KNOV 」

aman.esp
ass_m.esp
assum_n.esp
db_m.esp
db_mm.esp
dmce.esp
fman.esp
good_db.esp
good_val.esp
inf.esp
inode_get.esp
kb_m.esp
kco.esp
kvs.esp
kvs_win.esp
kvs_menu.esp
p_atms.esp
r_atms.esp
ram.esp
rctl.esp
rman.esp
rule_net.esp
rule_node.esp
var_node.esp
vinf.esp

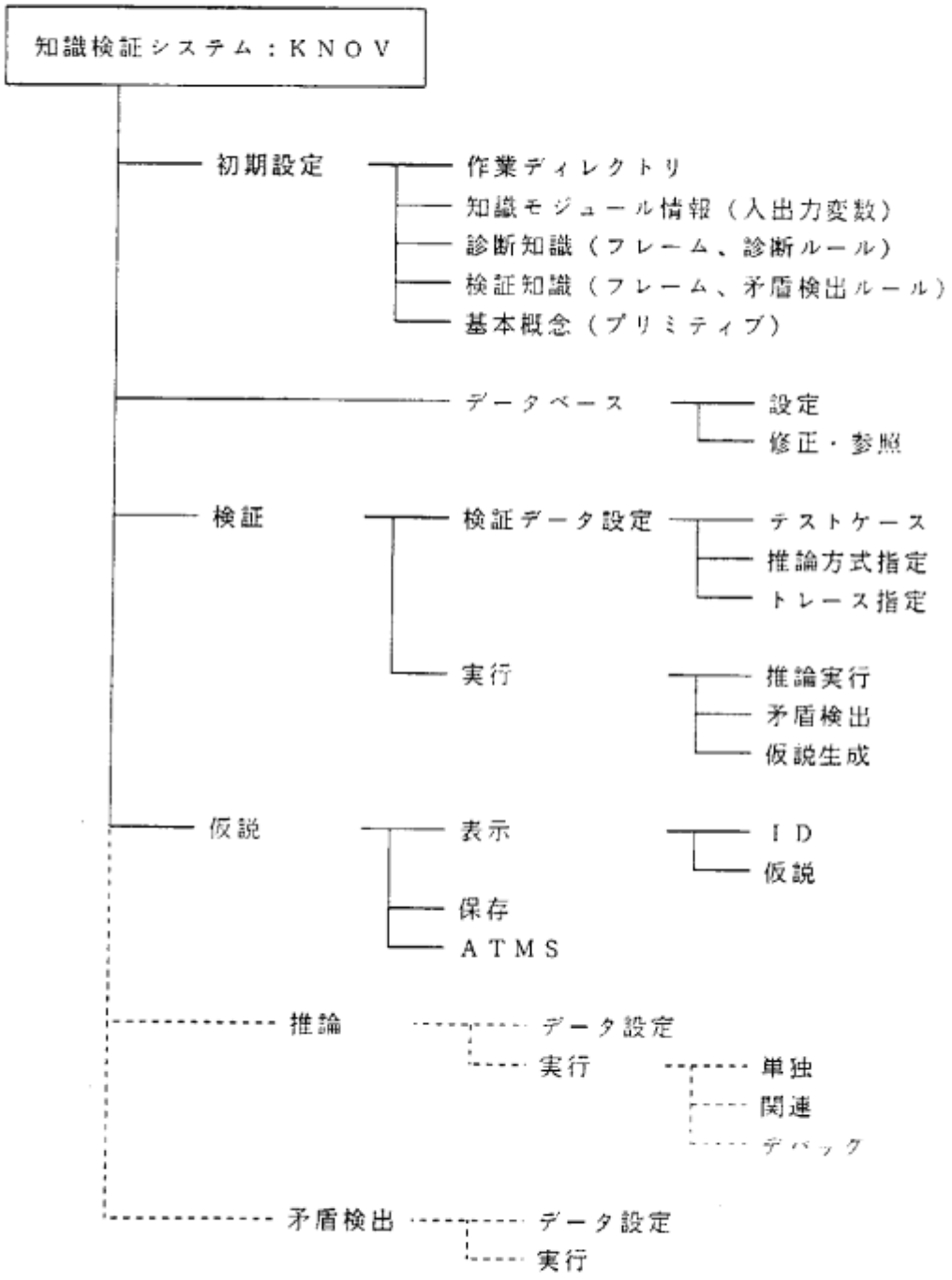
「 Apricot/0 」

ana.esp
atms.esp
con.esp
env.esp
in.esp
int.esp
net.esp
node.esp
ratms.esp
renv.esp
rin.esp
runion.esp
sol.esp
union.esp

4.2 操作方法

4.2.1 操作概要

知識検証システムの操作はウィンド画面を用いて対話形式で行う。以下に対話操作機能の体系を示す。



対話機能の個々の項目については以下に示す通りである。通常使用するのは(1)から(4)の機能であり、(5),(6)は診断ルール、矛盾検出ルールの詳細なデバッグを行う場合に用いる。

(1)初期設定：知識検証を行うために必要な情報の初期設定を行う。
(2)データベース：
　　データベースの設定と入力したデータベースの修正・確認を行う。
(2)検証：　検証機能では、以下の手順で推論ルールを検証する。
　　1.診断ルールを実行する。
　　2.推論結果の矛盾検出を行う。
　　3.矛盾ルールがあれば、仮説生成・検証を行いルール修正を行う。
(3)仮説：　検証機能で作成した仮説を画面上に表示し、ファイルに保存する。
(4)推論：　診断ルールを単独で実行する。
(5)矛盾検出：矛盾検出ルールを単独で実行する。

知識検証システムの対話画面は入出力画面とメニューに分けられる。
入出力画面はｐｍａｃｓ機能付きウィンドウを使用している。以下の５枚のウィンドウが重なりあった状態となっており、操作中の機能によって切り替えられる。2)～5)の画面は、表示終了後入力待状態になっているため、改行を行い1)の画面に戻してメインメニューの選択を行う。

1)メイン画面：システム全体のメッセージ、ガイダンスを表示する。
2)検証画面：　検証機能での入力、メッセージ、ガイダンスの出力を行う。
3)仮説画面：　仮説、ＡＴＭＳの表示機能での入力、メッセージ、ガイダンスの出力を行う。
4)データベース編集画面：
　　データベース編集機能での入力、メッセージ、ガイダンスの出力を行う。
5)推論、矛盾検出画面：
　　推論、矛盾検出機能での入力、メッセージ、ガイダンスの出力を行う。

メニューは、シングルコラムメニューとチョイスメニューを使用している。

a.シングルコラムメニュー：
　　マウスを項目上で移動するとboxマーカで囲まれる。左ボタンをクリックすると項目が選択される。
b.チョイスメニュー：
　　反転表示している項目が選択されている状態を示している。キー入力をする項目はクリックするとカーソルが表示される。改行を行うと入力が終了する。実行は、"do_it"をクリックして指示する。

4.2.2 操作方法

4.2.2.1 起動方法

システムメニューを呼び出し、"EXCV"をクリックする。以下に示すウィンドウ枠が現れるのでマウスを移動してウィンドウを表示する位置を決める。

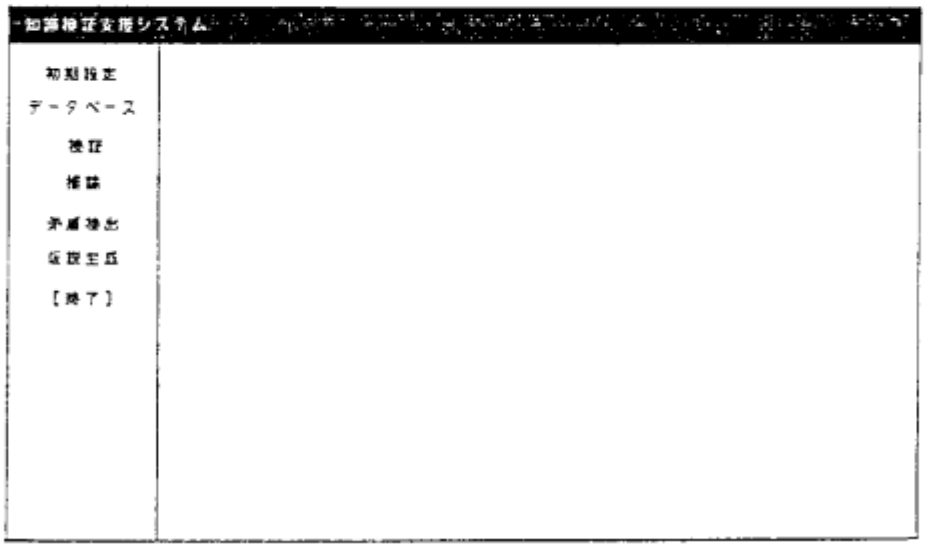


図4-3 初期画面

4.2.2.2 初期設定

検証に必要な情報の設定を行う。メインメニューの初期設定をクリックすると、初期設定用のサブメニューが表示される。

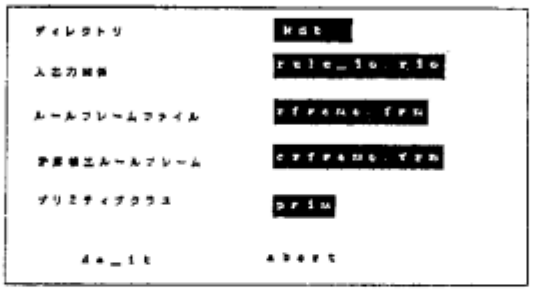


図4-4 初期設定サブメニュウ

サブメニューに表示されている以下の項目を設定する。

1)ディレクトリ：	作業ディレクトリ名
2)知識モジュール情報：	知識モジュール情報ファイル名
3)ルールフレーム：	診断ルールフレームファイル名
4)矛盾ルールフレーム：	矛盾検出ルールフレームファイル名
5)プリミティブクラス：	基本概念のクラス名

これらの項目を設定後do_itを指定すると初期化が行われ、設定結果が画面に表示される。

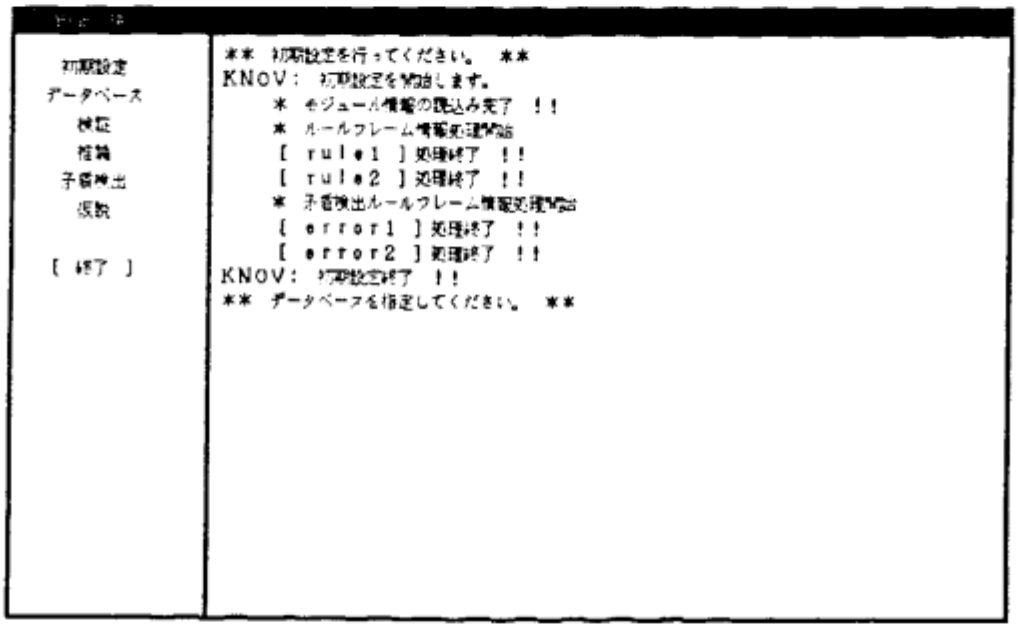


図4-5 初期設定経過表示画面

4.2.2.3　データベース

データベースの設定と変更を行う。メインメニュー上でデータベースをクリックすると設定・変更を選択するサブメニューが表示される。

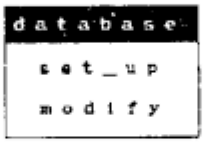


図4-6　機能選択サブメニュー

(1)設定

サブメニュー上で"set_up"をクリックすると設定メニューが表示される。

図4-7　データベース設定メニュー

メニュー上で以下の情報を入力し、"do_it"を指定するとデータベースの入力が行われる。

データベースファイル：データベースを記述したファイル名
補助情報クラス：　　補助情報クラス名

補助情報クラスはユーザによる機能拡張を行う場合に用いる。例えば推論過程のグラフィック表示機能の組み込む場合などに用いる。

(2)変更

データベースの変更を行う。この機能で提供している変更とは設定されたデータの変更であり、ファイル内のデータは変更しない。ファイル内のデータベースの変更はエディタで行う。

変更を選択すると、メインメニュー部分に以下に示す機能選択メニューが表示されるので、機能を選択する。

図4-8　データベース変更メニュー

機能を選択するとプロムプトが表示され入力待ちとなる。必要なデータを入力すると機能が実行される。以下にデータベースの変更、修正画面の例を示す。

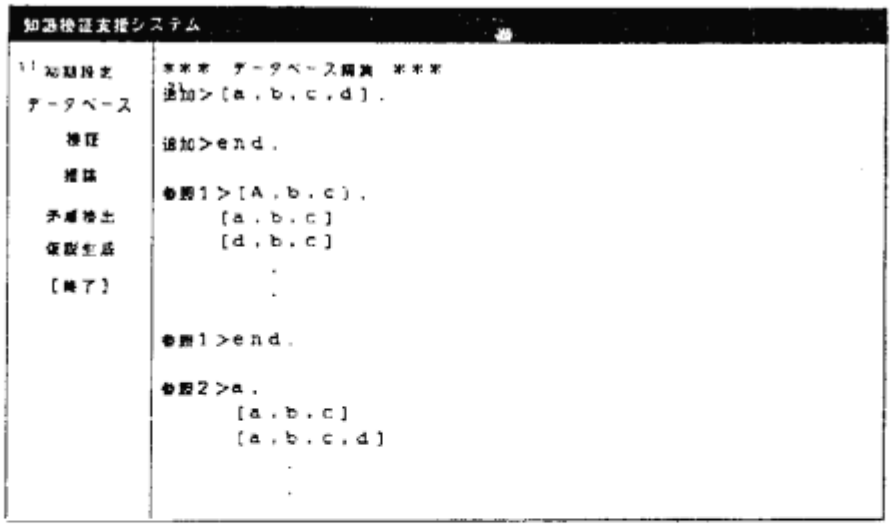


図4-9　データベースの変更画面

追加、修正、参照するデータは以下の形式で指定する。これは、データベースのファイル記述形式 db(Arg1,Arg2,....Argn) に対応している。

[Arg1,Arg2,....,Argn].

以下にデータベース変更における対話形式を示す。

表4-1　データベース変更の対話形式

機能	処理	対話形式
追加	データの追加を行う。	追加＞[arg1,arg2,....,argn]. 追加＞e.
削除	データの削除を行う。	削除＞[arg1,arg2,....,argn]. 削除＞e.
置換	データの置き換えを行う。	置換 NEW＞[arg1,arg2,....,argn]. OLD＞[arg1,arg2,....,argn].
参照1	リストによるデータの参照を行う。同一引き数数でマッチするデータを表示する	参照1＞[X,arg2,....,argn]. [a,arg2,....,argn] [b,arg2,....,argn] .....
参照2	キーによるデータの参照を行う。入力されたキーを含むデータを表示する。	参照2＞a. [a,arg2,....,argn] [arg1,a,....,argn] .....

4.2.2.4　検証

メインメニューで、検証をクリックすると検証設定メニューが表示される。メニュー上で検証で使用する情報を設定する。

図4-10　検証設定メニュー

メニューの指定項目は以下に示す通りである。

1)テストケース：　テストケースファイルを指定する。
2)推論手法：　推論を行う際の探査範囲を設定する。
　全探査　..ルールをすべて実行する。
　優先度順探査..指定された優先度まで推論を実行する。
　逐次探査　..採用されるルールがあれば推論を終わりにする。
3)最大優先度：　優先度順探査で推論する優先度の最大値を指定する。
4)トレース：　検証実行中の過程を表示するかしないかを指定する。

設定を行うと推論から矛盾検出まで自動的に実行する。

矛盾検出後、仮説生成を実行するか否かを問い合わせて来るので、仮説生成を行う場合には"y."を入力する。

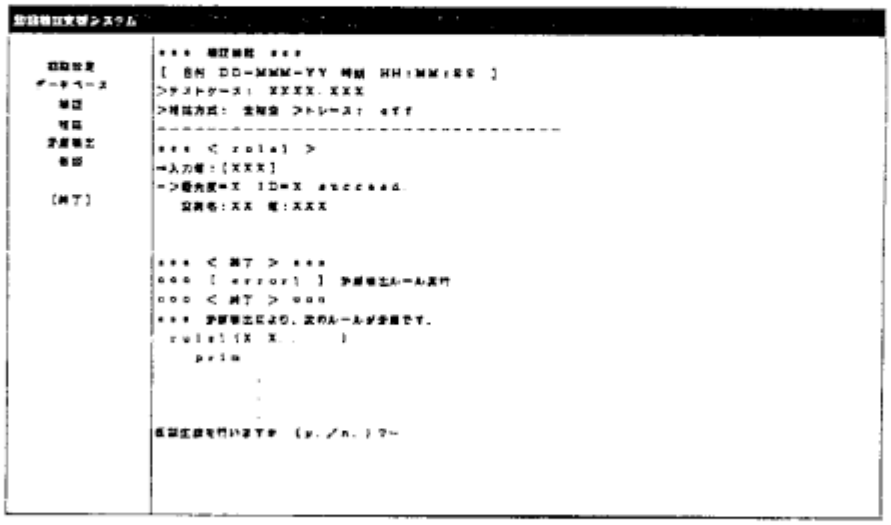


図4-11　検証実行表示例

4.2.2.5　仮説

生成した仮説を表示、保存する。以下のメニューより機能を選択する。

図4-12　仮説操作選択メニュー

（１）仮説表示..display
生成した仮説の表示を行う。以下の情報を表示する。

1)ルール名：　生成対象となったルール名。
2)開始ＩＤ：　仮説の識別子のナンバリング開始番号
3)全仮説数：　生成した仮説の総数
4)無矛盾仮説数：矛盾した結果を出さなかった仮説数
5)矛盾仮説数：　矛盾した結果を出した仮説数
6)未成功仮説数：実行中に失敗した仮説数
7)成功仮説数：　実行し、成功した仮説数
8)成功仮説数（同一結果）：
仮説生成対象となったルールと同じ結果を出した仮説数

詳細な情報は、識別子表示、仮説表示を選択して表示する。

1.識別子表示
上記の3)から8)の各項目を番号で選択し、
属する仮説の識別子を表示する。
2.仮説表示
識別子を指定し、仮説の内容を表示する。

以下に仮説表示画面の例を示す。

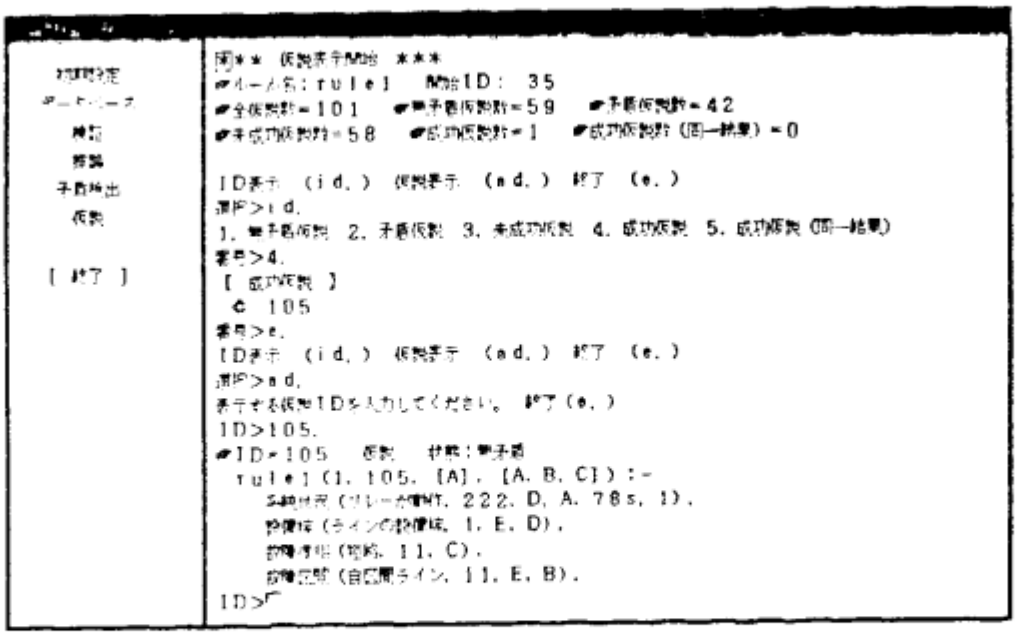


図4-13　仮説表示画面

（２）生成仮説保存..save

生成した仮説をファイルに保存する。以下の項目を設定し保存する。保存先ファイルは診断ルールフレームで指定されている診断ルールファイルとされる（4.3.2参照）。

1)ルール名：仮説のルール名

2)引数：　同一ルール名で異なる引数数のルールがある場合に引数数を指定する。

（３）ＡＴＭＳ表示

仮説生成・検証時に実行した全ての診断ルール（仮説を含む）の入力値、出力値を表示する。最初にＡＴＭＳで管理しているノード中どのルールのノードを表示するかを選択する。

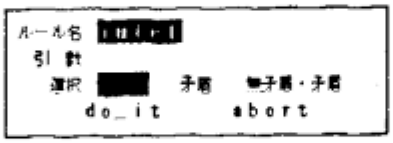


図4-14　ATMS表示選択メニュー

ＡＴＭＳ表示選択メニューの項目は以下に示す通りである。

1)ルール名：仮説のルール名を指定する。
2)引数　　：同一ルール名で、引数数が異なるルールがある場合に
　　引数数を指定する。

"do_it"の指定によりＡＴＭＳ情報（仮説を実行した際の、入力値、出力値）を表示する。
以下にＡＴＭＳ表示画面の例を示す。表示時に示される記号は、次の意味を持っている。

○　無矛盾を意味する
⊥　矛盾を意味する。

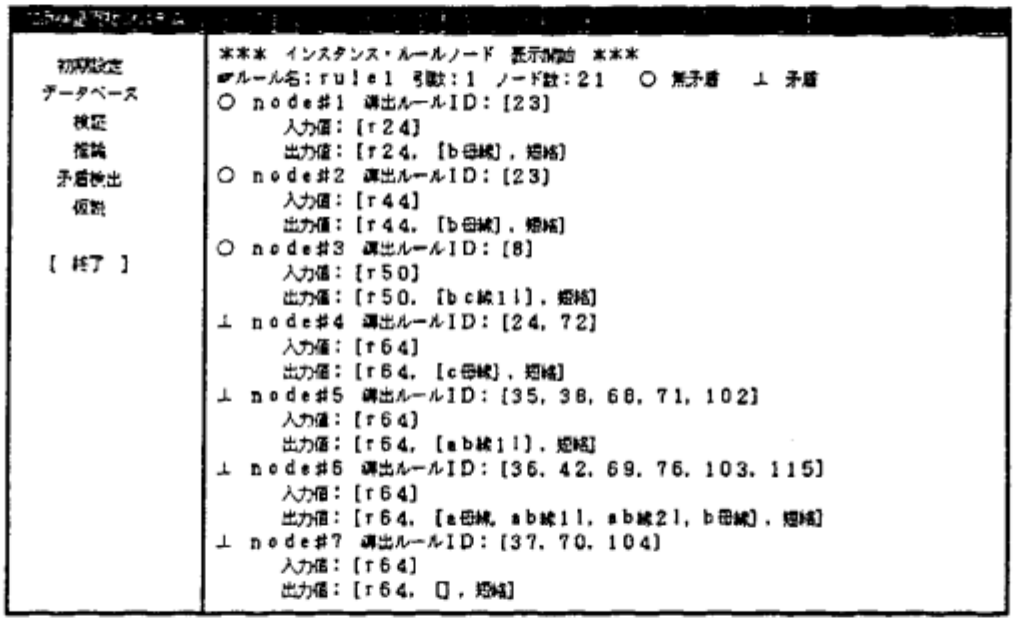


図4-15　仮説表示画面の例

4.2.2.6 終了

メニュー中の"[終了]"をクリックし、サブメニューの"yes"を選択する。

4.2.2.7　推論

推論機能のみを実行する。推論設定メニューで推論パラメータを設定する。

図4-16　推論設定メニュー

メニューの指定項目は以下に示す通りである。

1)ルール名：　推論を行うルールの名称を入力する。
2)引数：　同一ルール名で引数の異なるルールがある場合には実行したいルールの引数数を入力する。
3)推論手法：　探索の範囲を指定する。検証の推論手法と同じ。
4)優先度：　優先度順探査の優先度の最大値を指定する。
5)推論単位：
　1.単独　..ルール名で指定したルールのみを実行する。
　2.関連　..指定したルールが出力する変数を入力とするルールを走らせる。
　3.デバッグ..指定したルールの識別番号を入力し、ルールの中の1つだけを実行する。基本概念が成功すれば値がマッチした状態で基本概念を表示し、失敗すればfailと表示する。

推論設定メニュー中の"do_it"をクリックすることで推論が実行される。
推論を実行すると、ルールの実行に必要な入力変数の入力値を問い合わせて来るので以下に示す様にルールの入力変数に値を設定する。

入力例

```
A ?-abc.

B ?-def.
```

単独、関連指定の推論の実行結果は以下に示す様に、成功したルールの識別番号、優先度、入力値、出力値が表示される。

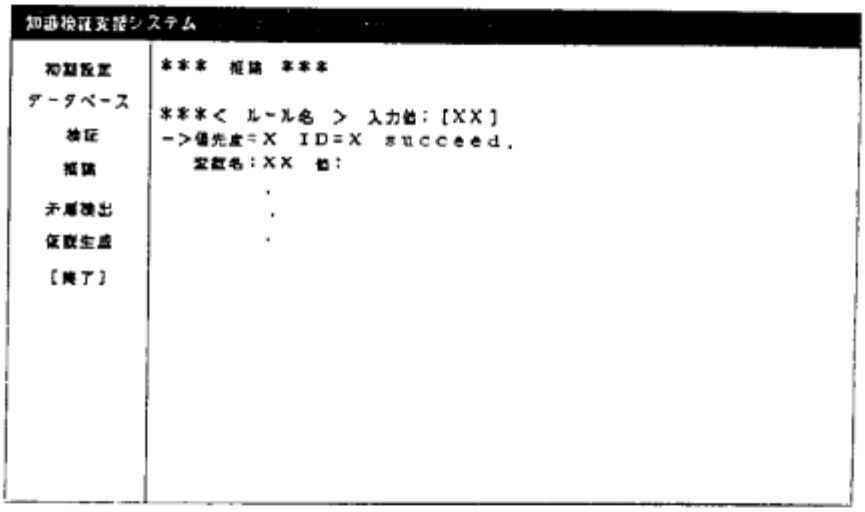


図4-17 推論実行表示画面（単独、関連）

デバッグモードで実行する場合には、実行するルールのルール識別番号（ID）を入力する。ルールを実行し、結果を表示したら、次に実行するルールの識別番号の入力待になる。入力の終了は"e."の入力とする。ルール実行に用いる入力変数の値はルール識別番号の入力が終了するまで同じ値を使用する。

入力例

```
ID>1.
```

デバッグ指定の推論実行結果は以下に示す様に、ルール中の成功した基本概念がその位

置と共に表示される。

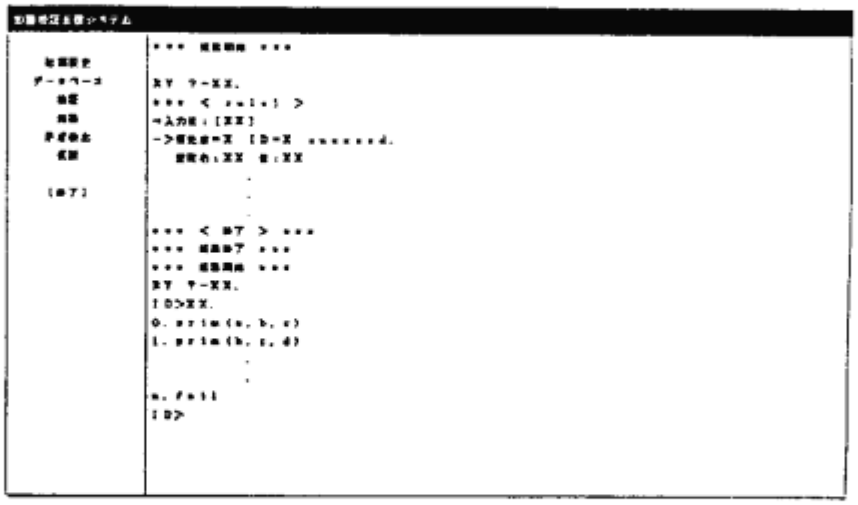


図4-18　デバッグモード表示画面

4.2.2.8　矛盾検出

矛盾検出を単独で実行する。矛盾検出パラメータは以下に示す矛盾検出メニューにより設定する。

図4-19　矛盾検出設定メニュー

メニューの指定項目は以下に示す通りである。

1)ルール名：実行する矛盾検出ルール名を指定する。
2)引数：　同一ルール名で、引数数の異なるルールがある場合には、引数数を入力する。
3)デバッグ：デバッグモードで実行するかを選択する。

メニュー中の"do_it"をクリックすることで矛盾検出定が実行される。
実行すると、矛盾検出ルールの入力変数に対する値を問い合わせて来るので、入力値の設定を行う。実行後矛盾と判断された診断ルールが表示される。以下に矛盾検出の対話画面の例を示す。

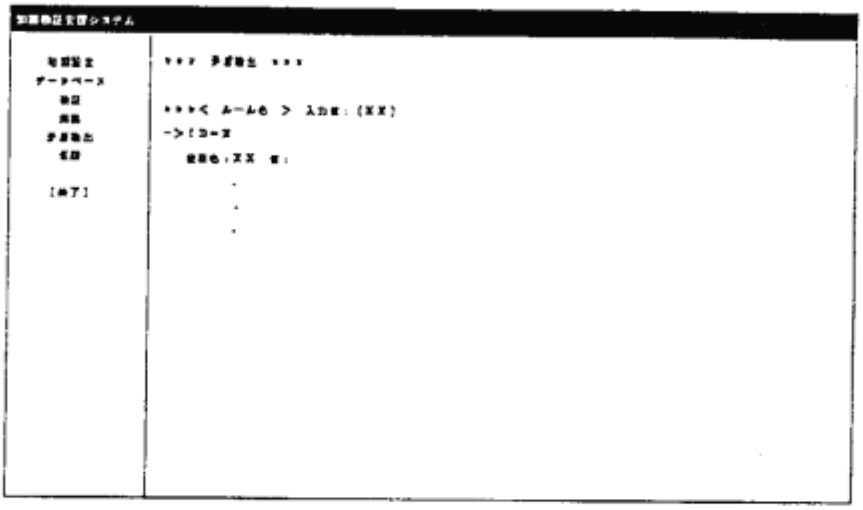


図4-20　矛盾検出表示例

4.3 知識及びデータ記述規則

本システムを起動する前に、予め次の知識およびデータを作成しておく必要がある。

4.3.1 知識モジュール情報

知識モジュール間の入出力関係情報を定義する。これはルールネットワークを作成する為に用いる。

ルール名（ルールタイプ，レベル，入力変数定義，出力変数定義）

ルール名　　：診断ルール名または矛盾検出ルール名
ルールタイプ　：ａまたはｃ　ａ＝診断ルール、ｃ＝矛盾検出ルール
レベル　　：同一ルールで入力変数の数が異なる時の推論優先度
　　値の小さい程優先度が高い
入力変数定義　：［入力変数，入力変数・・・］
出力変数定義　：［出力変数，出力変数・・・］

以下に記述例を示す。

```
(ex.)   rule1(a,1,[RY],[RY1,AREA,KIND]).
        rule2(a,1,[RY,RY1,AREA,KIND],[MAL_FUNC]).
        rule2(a,2,[RY,X],[Y]).
        error1(c,1,[RY,AREA,KIND],[RID]).
        error2(c,1,[RY,AREA,KIND,MAL_FUNC],[RID]).
```

（注）ルール２の場合入力変数が異なっている。この時何れのルールも実行可能な場合すなわち入力変数が定まっている場合、優先度の高いルールから先に実行される。

4.3.2　診断ルールフレーム

診断ルールフレームは、診断知識の枠組みを規定するものであり、推論及び仮説生成のベースとなる知識である。以下に示すフォーマットで記述する。

```
ｆｒａｍｅ（ﾙｰﾙ名，入力変数定義，出力変数定義，ﾌﾚｰﾑ定義，
            診断ﾙｰﾙﾌｧｲﾙ名，｛仮説生成知識ﾌｧｲﾙ名，仮説生成情報ﾌｧｲﾙ名｝）．

                                        {}　・・省略可
```

ﾙｰﾙ名	：ﾙｰﾙﾌﾚｰﾑ設定対象ﾙｰﾙ名（ｱﾄﾑ）
入力変数定義	：［ｼﾝﾎﾞﾙ名（ｽﾄﾘﾝｸﾞ），ｼﾝﾎﾞﾙ名　・・・・］
出力変数定義	：［ｼﾝﾎﾞﾙ名（ｽﾄﾘﾝｸﾞ），ｼﾝﾎﾞﾙ名　・・・・］
ﾌﾚｰﾑ定義	：［全体のﾌﾚｰﾑ］，［［条件部のﾌﾚｰﾑ］，［結論部のﾌﾚｰﾑ］］
診断ﾙｰﾙﾌｧｲﾙ名	：診断ﾙｰﾙﾌｧｲﾙ名（ｱﾄﾑ）　拡張子「.rule」固定
仮説生成知識ﾌｧｲﾙ名	：当該ﾌｧｲﾙ名（ｱﾄﾑ）　拡張子「.inf」固定
仮説生成情報ﾌｧｲﾙ名	：当該ﾌｧｲﾙ名（ｱﾄﾑ）　拡張子「.flg」固定

記述例を以下に示す。

```
(ex. )  frame(rule1,["RY"],["RY","AREA","KIND"],
                [r,s,u,y,j],[[r,s,u],[y,j]],rule1.assum,flag).

        frame(rule2,["RY1","RY","AREA","KIND"],["MAL_FUNC"],
                [e,d],[[e],[d]]).
```

4.3.3 矛盾検出ルールフレーム

矛盾検出ルールフレームは、矛盾検出ルールの枠組みを規定するものであり、矛盾検出のベースとなる知識である。以下に示すフォーマットで記述する。

```
ｆｒａｍｅ（ﾙｰﾙ名，入力変数定義，出力変数定義，ﾌﾚｰﾑ定義，
                                        矛盾検出ﾙｰﾙﾌｧｲﾙ名）．
```

ﾙｰﾙ名	：ﾙｰﾙﾌﾚｰﾑ設定対象ﾙｰﾙ名（ｱﾄﾑ）
入力変数定義	：［ｼﾝﾎﾞﾙ名（ｽﾄﾘﾝｸﾞ），ｼﾝﾎﾞﾙ名　・・・・］
出力変数定義	：［ｼﾝﾎﾞﾙ名（ｽﾄﾘﾝｸﾞ），ｼﾝﾎﾞﾙ名　・・・・］
ﾌﾚｰﾑ定義	：［全体のﾌﾚｰﾑ］，［［条件部のﾌﾚｰﾑ］，［結論部のﾌﾚｰﾑ］］
矛盾検出ﾙｰﾙﾌｧｲﾙ名	：矛盾検出ﾙｰﾙﾌｧｲﾙ名（ｱﾄﾑ）　拡張子「.crule」固定

記述例を以下に示す。

```
(ex. )  frame(error1,["RY","AREA","KIND"],["RID"],
                                'SSvoid','SSvoid',error1).
```

4.3.4 診断ルール

診断ルールの記述形式は、基本的にｐｒｏｌｏｇに準拠した形で記述する。但し、ＯＲ条件は、表現できない。以下に記述形式を示す。

```
ルール名（優先度，識別番号，［入力］，［出力］）：－
        プリミティブi　，
        プリミティブj　，
              ・
              ・
        プリミティブn　．
```

ルール名	：診断ルール名
優先度	：優先度推論時に参照される優先度
識別番号	：ルールの識別番号
入力	：［入力変数，入力変数・・・］
出力	：［出力変数，出力変数・・・］
プリミティブ	：条件及び結論の基本概念

記述例を以下に示す。

```
(ex.)   rule1(1,1,[RY],[RY,AREA,KIND]):-
            u(relay_on,222,SW,RY,'17g',D),
            s(receive_line,1,LINE,SW),
            u(parallel_line,111,LINE,PAIR),
            y('1LG',121,KIND),
            j(自区間,11,LINE,AREA).

        rule1(1,2,[RY],[RY,AREA,KIND]):-
            u(relay_on,222,SW,RY,'17s',D),
            s(receive_line,1,LINE,SW),
            u(parallel_line,111,LINE,PAIR),
            y(短絡,11,KIND),
            j(自区間,11,LINE,AREA).
```

4.3.5 プリミティブ

本システムを起動する前に、利用者はプリミティブ（基本概念）の一つ一つをメソッドとしたクラスを作成しカタログしなければならない。これはプリミティブ自身は、無矛盾であると考えるからである。また、ルールの実行を直接メソッド呼出することにより推論の高速化を図っている。以下にプリミティブクラスの記述形式を示す。

```
class ｸﾗｽ名 has

   attribute
        atms
        ;

   :ﾌﾟﾘﾐﾃｨﾌﾞ名(Class,DB,description,識別子,Arg1,Arg2･･･):-
             ﾎﾞﾃﾞｨ-部(任意)
             ;
                        .
                        .
local
                        .
                        .
end.
```

・スロット名atms･･･推論結果をアクセスするオブジェクトを保持
　本システムでは推論結果をＡＴＭＳに保持しており、ノードをアクセスするオブジェクトを提供している。利用しない場合でも記述が必要である。

・DB･･･データベースをアクセスするメソッド名:dbがあるオブジェクト
　推論側より第２引数にオブジェクトを引き渡す。

・description･･･プリミティブ（識別子）の説明記述

・識別子･･･プリミティブの識別子。説明記述に対応する。

・Arg･･･任意

以下に記述例を示す。

```
class  example  has
    attribute
       atms
       :
    :s(Obj,DB,line_side,1,LINE,SW):-
          :db(DB,type,LINE,line,STAT),
          :db(DB,receive,LINE,LIST1),
          :db(DB,supply,LINE,LIST2),
          ( :a(Obj,DB,....,1,SW,LIST1);
            :a(Obj,DB,....,1,SW,LIST2) )
          ;
               .
               .
               .
local
               .
               .
               .
end.
```

4.3.6　データベース

データベースの定義フォーマットを図以下に示す。

ｄｂ（データ1，データ2，データ3・・・・データn）　または ［データ1，データ2，データ3・・・・データn］ n・・最大１０個

データ：任意。

以下に記述例を示す。

(ex.)

```
db(type,AB線11,line,on).
db(type,AB線21,line,off).
db(type,s1,switch,on).
db(type,s5,switch,off).
db(receive,AB線11,[s1]).
db(receive,s1,[a_bus]).
db(supply,a_bus,[s1,s3]).
db(supply,s1,[AB線11]).
db(sw,s0,on,on).
db(sw,s5,on,off).
db(relay,'s0_51s','51s',s0,1,off).
db(relay,'s5_51s','51s',s5,1,on).
```

4.3.7　テストケース

テストケースは、検証の為の情報であり、以下データの定義を行う。

1. 判断基準、即ち時期状況に対応する正しい診断結果（事故原因）
   矛盾検出時に利用する
2. 事故状況を定義するためのデータベースへの追加、修正
3. 検証指定。診断推論または検証推論を開始する指定。
   入力変数の値を指定して、それを入力とするルールの実行を開始させる。

これらについて、以下に記述形式を示す。

```
[ｄｂ（ｔｅｓｔ，テスト名，［入力値］，［出力値］）]

 ｄｂ（データ1，データ2・・・）[
[ｄｂ（データ1，データ2・・・），ｄｂ（データ1'，データ2'・・・）]

ｔｅｓｔ（［入力変数名］，［［入力値1］，［入力値2］・・・］）
```

入力変数名：入出力関係で定義した入力変数名（ｽﾄﾘﾝｸﾞ）
入力値　　：任意（事故現象など）
出力値　　：任意（事故原因など）

（注）検証指定において入力変数名及び入力値を複数指定した場合、実行対象となるルールが複数存在する場合が生じる。この時入力値が同一でも複数回推論を行う。
例えば以下の検証指定の場合、入力変数"B"を持つルールrule2に対してb1を入力値とする推論を二度実行する。

```
test(["A","B"],[[a1,b1],[a2,b1],[a3]])

rule1(...[A,B],[X]).
rule2(...[B],[Y]).
```

ｄｂ（・・・）の記述は必要が無ければ省略可能である。

以下に記述例を示す。

```
(ex.)
      [db(test,1,[r51],[[bc線11,[c母線,cd線11,cd線21],短絡])].
      [db(test,2,[r84],[[c母線],短絡)].

      [db(relay,r51,'51s',s5,1,off), db(relay,r51,'51s',s5,1,on)].
      [db(relay,r84,'78s',s8,-1,off),db(relay,r84,'78s',s8,-1,on)].
      [db(type,c母線,bus,on),        db(type,c母線,bus,off)].
      [db(type,d母線,bus,on),        db(type,d母線,bus,off)].
      [db(type,cd線11,line,on),      db(type,cd線11,line,off)].
      [db(type,cd線21,line,on),      db(type,cd線21,line,off)].
      [db(type,bc線11,line,on),      db(type,bc線11,line,off)].

      test(["RY"],[[r51],[r84]]).
                        .
                        .
                        .          データベースの追加・修正
                        .            (事故状況の再設定)
                        .
      test(["RY"],[[r10],[r61],[r81]]).
```

（注）最初のtestの指定で変数"RY"の値をr51として推論を開始させる。
その後変数"RY"の値をr84として推論が行われる。
test指定による一連の推論の終了後、データベースは変更前の値に戻される。

データベースの追加・修正により次の事故状況を設定し二回目のtest指定で、再度推論を開始する。

4.3.8 矛盾検出ルール

矛盾検出ルールの記述形式は、基本的にｐｒｏｌｏｇに準拠した形で記述する。但し、ＯＲ条件は、表現できない。以下に記述形式を示す。

```
ルール名（優先度，識別番号，対象ルール名，［入力］，［出力］）：
                プリミティブi ，
                プリミティブj ，
                      ・
                      ・
                プリミティブn ．
```

ルール名	：矛盾検出ルール名
優先度	：優先度推論時に参照される優先度
識別番号	：ルールの識別番号
対象ルール名	：矛盾対象ルール名
入力	：「入力変数，入力変数・・・」
出力	：［出力変数，出力変数・・・］
プリミティブ	：条件及び結論の基本概念

以下に記述例を示す。

```
(ex.)  error1(1,2000,rule1,[RY,AREA,KIND],[2000]):-
            u(test_case_io,3,[RY],[Y,Z]),
            a(equal,11,KIND,Z),
            not(a(list_equal,31,AREA,Y)).
       error1(1,2001,rule1,[RY,AREA,KIND],[2001]):-
            u(test_case_io,3,[RY],[Y,Z]),
            a(list_equal,31,AREA,Y),
            a(not_equal ,12,KIND,Z).
```

4.3.9 仮説生成知識

仮説生成知識は、仮説を生成するための基盤となる知識であり、診断対象領域の構造を階層関係として定義する。本知識をもとに仮説生成を行い診断ルールを作り出す。以下に定義フォーマットを示す。

```
def(p_object,[c_object... ])
```

p_object：上位概念名（項目について定義しようとする実体）
p_object：下位概念名（項目について定義しようとする実体）

（注）概念名は仮説生成知識内で重複して定義してはならない。
最上位概念名は"top_def"とする。

4.3.10　仮説生成情報

仮説生成情報には以下のものがある。

1. 基本概念の分類(p_group)
   基本概念（プリミティブ）はその内容によりいくつかにグループ化される。
2. 参照(ref)
   仮説生成を行う過程で必要となる情報であり、仮説生成知識とプリミティブとの対応関係の定義と、仮説生成知識上の検索戦略の情報を持つ。
3. 戦略(strategy)
   仮説生成を行う場合、仮説生成知識からその対応するプリミティブを組み合わせて診断ルールを生成する。その際に対象とする仮説生成知識の範囲を規定する。これは、プリミティブグループ毎に指定する。

以下に、それぞれの記述形式を示す。

```
flag(p_group,[グループ名、述語名、[識別番号]])
```

グループ名：プリミティブのグループ名
述語名：　プリミティブの述語名またはall
　　all：任意の述語
識別番号：　allまたはプリミティブ番号,...
　　all：指定した述語名のプリミティブ全てを含む場合
　　プリミティブ番号：プリミティブ番号を指定する

```
flag(ref,[object,[ﾌﾟﾘﾐﾃｨﾌﾞ頭部]])
```

object　　：仮説生成知識defの第1項
ﾌﾟﾘﾐﾃｨﾌﾞ頭部：対応するﾌﾟﾘﾐﾃｨﾌﾞの述語頭部

flag(strategy,[ﾌﾟﾘﾐﾃｨﾌﾞｸﾞﾙｰﾌﾟ名,戦略,[項目,object,項目値]])

ﾌﾟﾘﾐﾃｨﾌﾞｸﾞﾙｰﾌﾟ名：ﾌﾟﾘﾐﾃｨﾌﾞのｸﾞﾙｰﾌﾟ名

戦略　　：戦略の指定。１または２。

１：全ての仮説生成知識の組み合わせを用いてルール（仮説）を生成する。

２：下位概念と上位概念の１レベルと全ての並列概念の範囲でルール（仮説）を生成する。

[項目,object,項目値]：

ﾌﾟﾘﾐﾃｨﾌﾞｸﾞﾙｰﾌﾟ名で指定した仮説生成を、仮説生成知識のobject以下の知識を用いて行う。

[5] まとめ

知識検証システムは、知識ベースの中の矛盾知識を検出し、それを修正する問題解決機構と、矛盾知識に関する実行情報を管理し知識ベースの整合性を保つ知識管理機構で構成される。問題解決機構は矛盾知識の検出、それを修正するための仮説（知識候補案）の生成と検証を行う。このため検証の対象となる診断知識を推論し、矛盾定義知識（テストケース）、矛盾検出知識、仮説生成知識を用いて、矛盾検出、仮説生成、仮説検証のサイクルを繰り返す。この機能を実現する構成として矛盾検出、仮説生成部分を知識化し、問題領域毎に適した知識検証システムを実現できるようにした。知識管理機構に関してはＡＴＭＳを用いて矛盾知識をネットワークとして管理し仮説生成、仮説検証を効率よく行う方式を開発した。この方式は、知識の矛盾状態を集合として管理しているために、複数の状況における知識検証処理すなわち複数のテストケースによる知識ベースの確認・修正を一度にまとめて行う事が可能となった。これらをまとめると以下の点に集約される。

(1)矛盾検出、仮説生成の自動化による検証手続きの省力化

矛盾検出、仮説生成知識さえ与えておけば、システムが自動的に知識修正の全部の可能性をチェックし、知識ベースの修正を行う。このため人間が検証を行う場合の様に、先入観による思いこみや、不注意によるチェック漏れは生じない。

(2)テストケース群によるテスト

複数のテストケースにより知識ベースの検証が機械的にできる。これはＡＴＭＳにより矛盾知識を集合として管理するためである。人間による場合にはテスト戦略を立てて逐次的に矛盾部分の検出、修正、確認を行っているが、複数の修正案を集合として同時に検証ができる。従って「ある状況では正しいルールが、別の状況では誤った診断を行う」様なケースも矛盾無く管理ができる。

(3)検証知識の分離

知識フレーム、矛盾検出知識、仮説生成知識を、対象とする問題領域毎に設定できる構成となっており、問題領域に応じた知識検証システムとする事ができる。

知識検証方式の評価については、電気系統故障診断エキスパートシステムおよび、計算機システム復旧支援エキスパートシステムに適用し有効性を示した。

本研究の成果をベースにし、さらに実用に耐え得る知識検証ツールへ発展させるためには、以下の検討及び機能拡張を行う必要がある。

(1)検証知識の獲得

知識フレーム、矛盾検出知識、仮説生成知識などは、知識獲得・検証のためのメタ知識であり、一般の診断知識よりも高度な内容を持つ。これらの知識の抽出・整理・定式化をどのように行えば良いか。

(2)仮説生成の多様化

仮説生成知識に基づいた、ルールを構成する個々の基本概念の具体化、抽象化だけでなく基本概念についての省略、追加、順序の組み替えによるルール生成、推論時の競合解消のための推論戦略の修正・変更などが考えられる。またそのための仮説生成知識の拡張を行う必要が有る。

謝辞

本研究は第五世代コンピュータプロジェクトの一環として（財）新世代コンピュータ技術開発機構（ＩＣＯＴ）からの委託により行ったものである。知識検証システムの開発に際してご指導いただいたＩＣＯＴ第５研究室 生駒憲治室長および滝寛和主任研究員をはじめとする第五研究室の皆様に感謝致します。

参考文献

[太田 89]太田好彦，井上克巳，仮説推論システムＡＰＲＩＣＯＴ／０ユーザズ・マニュアル，ICOT TM-0676(1989).

[藤原 88]藤原遠，井上克巳，ＥＳＰによる仮説推論機構－ＡＳＴＲＯＮ－，ICOT TM-0587(1988).

[飯島 88]飯島勝美，井上克巳，ＥＳＰによるＡＴＭＳ－第１版－，ICOT TM-0467(1988).

[de Kleer 87]J.de Kleer, B.C.Williams, "Diagnosting Multiple Faults", Artificial Intelligence, Vol.32, pp.97-130(1987).

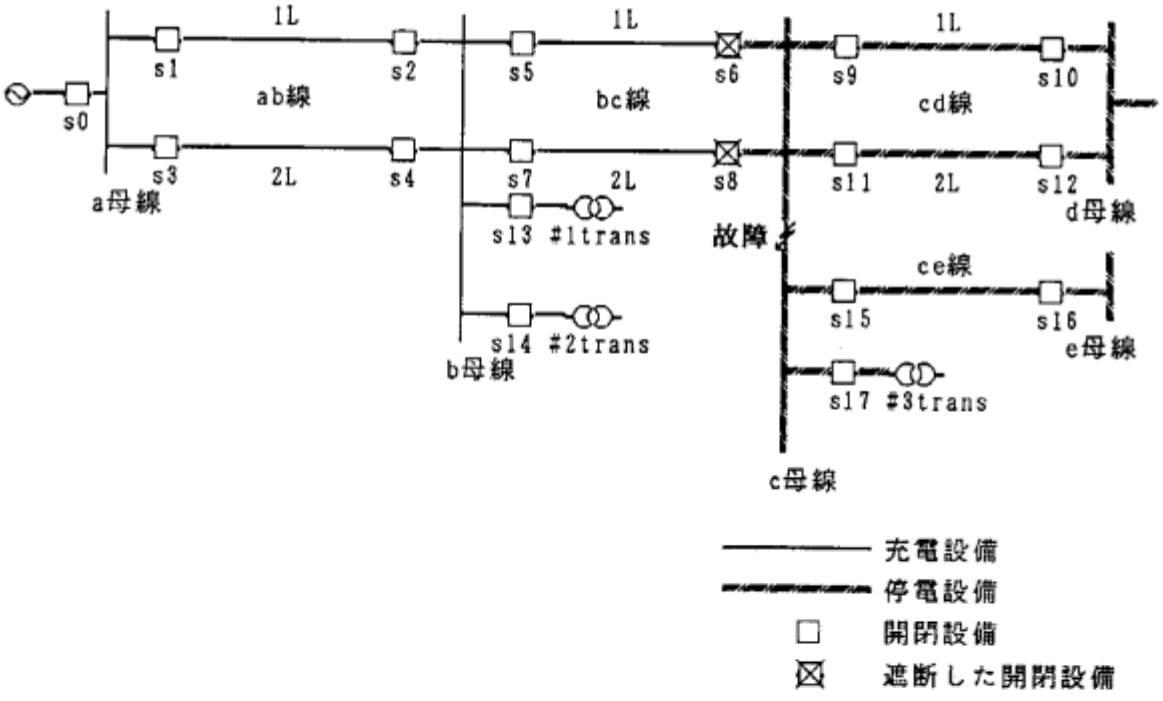
[de Kleer 86]J.de Kleer, "An assumption-based TMS", Artificial Intelligence, Vol.28, pp.127-162(1986).

付録-A　電気系統設備診断への適用

A.1　解説

A.1.1　電気系統故障診断

電気系統とは、様々な産業分野の工場などにおける構内受配電設備や、遠隔地間の送電／変電設備を意味する。このような系統設備には、通常、故障発生時に、故障を検知し、事故の波及を防ぐために設備を接続している開閉設備を遮断する、系統用保護リレー装置が設置されている。従って、電気系統で故障が発生した場合は、系統用保護リレーが動作し、その動作状況は電気系統監視用の計算機にデータとして取り込むことが可能である。このリレー動作状況と電気系統の状況を用いて、故障状況の解析をしようとするのが電気系統故障診断であり、解析のための知識を知識ベース化して、故障診断エキスパートシステムを形成する。電気系統故障例を図A-1に示す。図A-1の例は、c母線で故障が発生し、s6とs8の開閉設備が、各々に配置されている系統用保護リレーにより遮断され、c母線より下流の設備が全て停電になっている事を示す。



図A-1　電気系統故障例

電気系統故障診断エキスパートシステムは、故障状況として、動作リレー情報、しゃ断開閉設備情報、設備の充停電情報等から診断を行い、故障設備と故障様相（原因）を求める。

A.1.2　診断知識

診断ルール（フレーム）

(1)故障候補抽出ルール(rule1)

系統状況より系統用保護リレーの動作原理を用いて故障様相と故障設備を判断するものであり、個々のルールに優先度をつけルールの採用順序を定めているので、優先度順推論方式を用いる。

```
frame(rule1,['RY'],['AREA','KIND'],...,
            [[リレー動作,設備端,系統状況],[故障様相,故障区間]])
```

RY：　動作リレー
AREA：故障区間
KIND：故障様相

(2)誤動作リレー抽出ルール(rule2)

系統用保護リレーの誤動作を診断する。このルールは、個々のルール内容が独立であり、採用順序による採用ルールの違いは起こらないので、任意順の推論方式を用いる。

```
frame(rule2,['RY1','RY','AREA','KIND'],['MAL_FUNC'],...,[[e],[d]])
```

RY1：誤動作であるか否かをチェックするリレー
RY：　動作リレー（RY≠RY1）
AREA：故障区間
KIND：故障様相
MAL_FUNC：誤動作リレー

基本概念（プリミティブグループ）

```
flag(p_group,[設備端,設備端,all])              ...開閉設備設置位置
flag(p_group,[故障様相,故障様相,all])
flag(p_group,[故障区間,故障区間,all])
flag(p_group,[系統状況,系統状況,all])
flag(p_group,[リレー動作,系統状況,[222]])
flag(p_group,[操作,操作,all])
flag(p_group,[比較,操作,[11]])
flag(p_group,[リレー種別,リレー種別,all])
flag(p_group,[e,all,all])                      ...任意
flag(p_group,[d,操作,[12]])                    ...不等
```

データベース（系統構成定義形式）

電気系統の構成を述語'db'で定義する。第1引数が定義内容のキーワードになっており、それぞれの定義内容に応じて引数を設定する。

表A-1　電気系統構成述語定義

述語フォーマット	意味
db(type, SETUBI, TYPE, STAT)	SETUBI　系統設備名 TYPE　系統設備の型(line, trans, bus, switch STAT　設備が充電か停電か(on, off)
db(receive, SETUBI, LIST)	SETUBI　系統設備名 LIST　系統設備リスト
db(supply, SETUBI, LIST)	SETUBI　系統設備名 LIST　系統設備リスト
db(sw, SW, BEFORE, AFTER)	SW　開閉機器名 BEFORE　事故前の開閉状況　on/off AFTER　事故後の開閉状況　on/off
db(bus-tie, SW, BUS1, BUS2)	SW　ブスタイである開閉設備名 BUS1　複母線の母線名 BUS2　複母線のもう一方の母線名
db(trans, TRANS, PRIME, SECOND, THIRD)	TRANS　変圧器名 PRIME　一次側開閉設備名 SECOND　二次側開閉設備名 THIRD　三次側開閉設備名　ない時は"no"
db(relay, RY, DEVICE, SW, DIR, STAT)	RY　保護リレー名 DEVICE　リレーのタイプ　50s, 51s, 67g, etc. SW　保護リレーが設置されている開閉設備 DIR　保護リレーの保護方向(default=1) 67g, 78, 87については、'1'または'-1' STAT　保護リレーの動作状況 動作=on　動作していない=off
db(cb_list, LIST)	LIST　事故で遮断したＣＢリスト
db(ry_list, LIST)	LIST　事故で動作したリレーリスト
db(parallel, LINE1, LINE2)	LINE1　平行回線運用をするライン LINE2　LINE1のペアとなるライン
db(label, X)	X　推論毎に与えられる識別子

A.1.3　矛盾検出知識

(1)rule1、rule2共通

　　推論結果がテストケースで設定している結果と異なる　　（識別番号=2005～2007）

(2)rule1用

　　推論結果がテストケースで設定している結果と異なる　　（識別番号=2000,2001）

　　保護区間が充電している。　　（識別番号=2002）

　　短絡用のリレーが動作しているのに故障様相が地絡である（識別番号=2003）

　　地絡用のリレーが動作しているのに故障様相が短絡である（識別番号=2004）

(3)rule2用

　　推論結果がテストケースで設定している結果と異なる　　（識別番号=2008）

A.1.4　仮説生成

(1)仮説生成知識（部分）

```
power_system --- facility          --- side        --- line_side
 電気系統     |   系統設備          |   開閉器位置 +- bus_side
              |                     |               +- trans_side
              |                     |               +- switch_side
              |                     +- state       --- line_state
              |                     |   状態        +- bus_state
              |                     |               +- trans_state
              |                     |               +- switch_state
              |                     +-connection   --- line_connection
              |                        接続         +- bus_connection
              |                                     +- trans_connection
              |                                     +- switch_connection
              |
              +- protect_facility --- relay       --- relay_state       --- r_name
                  保護設備             リレー       |   リレー状態        +- device
                                                    |                     +- time
                                                    |                     +- switch_name
                                                    |                     +- action
                                                    +- relay_class        --- princple_class
                                                    |   リレー種類          +- usage_class
                                                    |                     +- kind_class
                                                    +- protect_area       --- ' 自区間'
                                                    |   保護区間            +- ' 3区間'
                                                    |                     +- ' 不明'
                                                    +- relay_kind         --- short
                                                        故障様相            +- ground
                                                                          +- dansen
                                                                          +- unknown
```

(2)仮説生成情報

診断ルールのif部に記述する条件項目（リレー動作、設備端、系統状況、操作）については、詳細化方向・並列概念・上位概念の各1レベルの項目を仮説生成の対象とする。then部に記述する結果項目（故障様相、故障区間）については全項目を仮説生成の対象とする。

A.1.5　テストケース

検証内容

系統保護リレーによる故障候補抽出ルール(rule1)と誤動作リレー検出ルール(rule2)の検証のために一般的には以下のテストを行う。

・rule1の検証：個々のリレーデバイスに対するルールの確認
　　複数のリレーが動作している場合の相互関係の確認
・rule2の検証：誤動作リレー判断の漏れ

本例では電気系統故障診断用の知識のうち、系統用保護リレーのデバイス'78s'に関するrule1知識の不備を検証操作によって修正する。

このため以下の矛盾ルールを知識に加える。

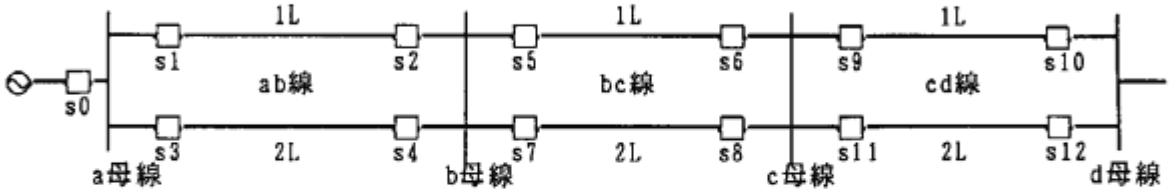
```
rule1(1,24,[RY],[AREA,KIND]):-
    系統状況(リレーが動作,222,SW,RY,'78s',1),
α →設備端(母線の設備端,2,BUS,SW),
    故障様相(短絡11,KIND),
β →故障区間(自区間母線,12,BUS,AREA).
```

検証後の修正目標となる知識は以下に示す通りである。
（rule1の第2引数のXXは、仮説生成時に決められる識別番号である）

```
rule1(1,XX,[RY],[AREA,KIND]):-
    系統状況(リレーが動作,222,SW,RY,'78s',1),
α1→設備端(ラインの設備端,1,LINE,SW),
    故障様相(短絡11,KIND),
β1→故障区間(自区間ライン,11,LINE,AREA).
```

この知識の修正は、並列概念による修正であり、αの部分を並列にα_1に修正し、それに対応してβの部分をβ_1に修正する。

テスト系統



図A-2　テスト電気系統

表A-2　系統用保護リレー設置状況

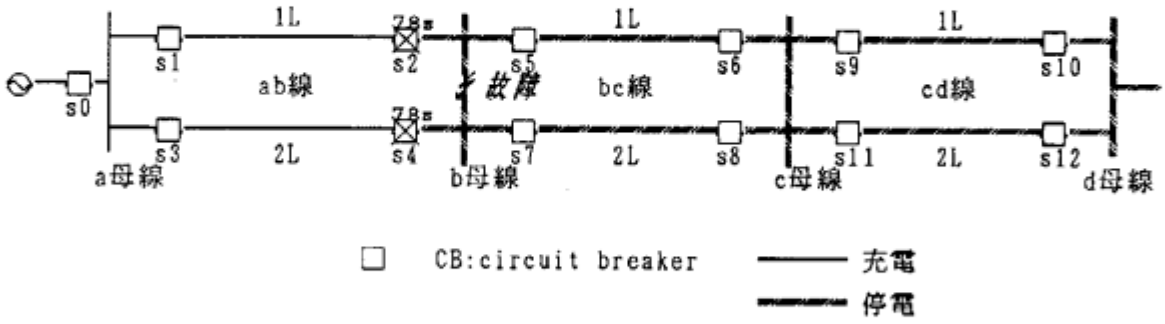
開閉設備 ＼ リレータイプ	50s	51s	78s
s0		r01	
s1	r10	r11	
s2	r20	r21	r24
s3	r30	r31	
s4	r40	r41	r44
s5	r50	r51	
s6	r60	r61	r64
s7	r70	r71	
s8	r80	r81	r84
s9	r90	r91	
s10	r100	r101	
s11	r110	r111	
s12	r120	r121	

（"rXXX"はリレー名）

テストデータ

(1)テストデータ1

①故障状況...母線で短絡事故が起きた場合



しゃ断ＣＢ	動作リレー	動作リレーのタイプ
s2	r24	78s
s4	r44	78s

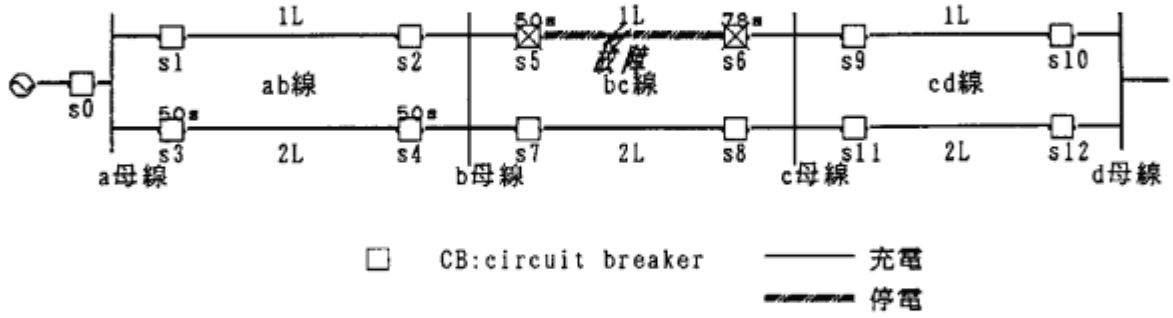
図A-3　テストデータ1

②故障判定結果

	動作リレー	故障区間	故障様相	誤動作リレー
総合判定	r24,r44	[b母線]	短絡	なし
rule1 適用結果	r24	[b母線]	短絡	
	r44	[b母線]	短絡	
rule2 適用結果				適用なし

(2)テストデータ２

①故障状況...ラインで短絡事故が起きた場合



□ CB:circuit breaker ——— 充電 ━━━ 停電

しゃ断ＣＢ	動作リレー	動作リレーのタイプ
s5	r50	50s
s6	r64	78s

図A-4　テストデータ２

②故障判定結果

	動作リレー	故障区間	故障様相	誤動作リレー
総合判定	r50,r64	[bc線1L]	短絡	なし
rule1 適用結果	r50	[bc線1L]	短絡	
	r64	[bc線1L]	短絡	
rule2 適用結果				適用なし

A.2　知識及びデータ記述

知識及びデータは次に示すファイルに定義されている。

知識構成

知識モジュール情報	rule_io.rio
診断知識フレーム	rf4.frm
矛盾検出ルールフレーム	cframe.frm

診断知識

診断ルール	trule4.rule
基本概念（プリミティブ）	prim.esp
データベース	db.db

検証知識

テストケース	tc4.tst
矛盾検出ルール	crl.crule
仮説生成知識	assum.inf
仮説生成情報	flag.flg

その他

補助情報表示	eg.esp(クラス名はeg)

```
tepsi916::>fd0>rule_io.rio

rule1(a, 0, [RY], [RY1, AREA, KIND]).
rule2(a, 1, [RY, RY1, AREA, KIND], [MAL_FUNC]).
error1(c, 1, [RY1, AREA, KIND], [RID]).
error2(c, 0, [RY, AREA, KIND, MAL_FUNC], [RID]).
```

```
tepsi916::>fd0>rf4.frm

frame(rule1, ["RY"], ["RY", "AREA", "KIND"],
          [リレー動作, 設備端, 系統状況, 故障復旧, 故障区間],
          [[リレー動作, 設備端, 系統状況], [故障復旧, 故障区間]],
          rule4, assum, flag).
frame(rule2, ["RY1", "RY", "AREA", "KIND"], ["MAL_FUNC"],
          [e, d], [[e], [d]], rule2).
```

```
tepsi916::>fd0>cframe.frm

frame(error1, ["RY", "AREA", "KIND"], ["RID"], '$$void', '$$void', cr1).
frame(error2, ["RY", "AREA", "KIND", "MAL_FUNC"], ["RID"],
          '$$void', '$$void', cr2).
```

```
teps1916::>fd0>trule4.rule

rule1(1, 1, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '17g', D),
      設備端(ラインの設備端, 1, LINE, SW),
      系統状況(平行回線運用, 111, LINE, PAIR),
      故障様相('1相地絡', 121, KIND),
      故障区間(自区間ライン, 11, LINE, AREA).
rule1(1, 2, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '17s', D),
      設備端(ラインの設備端, 1, LINE, SW),
      系統状況(平行回線運用, 111, LINE, PAIR),
      故障様相(短絡, 11, KIND),
      故障区間(自区間ライン, 11, LINE, AREA).
rule1(1, 3, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '27', D),
      故障様相(不明, 14, KIND),
      故障区間(故障区間不明, 3, AREA).
rule1(1, 4, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '44s1', D),
      設備端(ラインの受電端, 11, LINE, SW),
      故障様相(短絡, 11, KIND),
      故障区間(自区間ライン, 11, LINE, AREA).
rule1(1, 5, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '44s2', D),
      設備端(ラインの受電端, 11, LINE, SW),
      故障様相(短絡, 11, KIND),
      故障区間(ラインの受電端から下流に3区間, 211, LINE, AREA).
rule1(1, 6, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '50g', D),
      設備端(ラインの設備端, 1, LINE, SW),
      系統状況(平行回線運用, 111, LINE, PAIR),
      故障様相('1相地絡', 121, KIND),
      故障区間(自区間ライン, 11, LINE, AREA).
rule1(2, 7, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '50g', D),
      設備端(ラインの受電端, 11, LINE, SW),
      故障様相('1相地絡', 121, KIND),
      故障区間(ラインの受電端から下流に3区間, 211, LINE, AREA).
rule1(1, 8, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '50s', D),
      設備端(ラインの受電端, 11, LINE, SW),
      故障様相(短絡, 11, KIND),
      故障区間(自区間ライン, 11, LINE, AREA).
rule1(1, 10, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '51s', D),
      設備端(ラインの受電端, 11, LINE, SW),
      故障様相(短絡, 11, KIND),
      故障区間(ラインの受電端から下流に3区間, 211, LINE, AREA).
rule1(1, 11, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '51s', D),
      設備端(ラインの送電端, 12, LINE, SW),
      系統状況(対向端の開閉設備, 1213, LINE, SW, SW1),
      系統状況(開閉設備は事故遮断, 2111, SW1),
      故障様相(短絡, 11, KIND),
      故障区間(自区間ライン, 11, LINE, AREA).
rule1(1, 12, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '51s', D),
      設備端(ラインの送電端, 12, LINE, SW),
      not(系統状況(平行回線運用, 111, LINE, PAIR)),
      故障様相(短絡, 11, KIND),
      故障区間(ラインの送電端から下流に3区間, 212, LINE, AREA).
```

```
rule1(1, 13, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '51str', D),
      設備端(変圧器の1次側, 31, TRANS, SW),
      故障様相(短絡, 11, KIND),
      故障区間(自区間変圧器, 13, TRANS, AREA).
rule1(1, 14, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '51str', D),
      設備端(変圧器の2次側, 32, TRANS, SW),
      故障様相(短絡, 11, KIND),
      故障区間(自区間変圧器, 13, TRANS, AREA).
rule1(1, 15, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '63q', D),
      設備端(変圧器の設備端, 3, TRANS, SW),
      故障様相(不明, 14, KIND),
      故障区間(自区間変圧器, 13, TRANS, AREA).
rule1(3, 16, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '64b', D),
      設備端(変圧器の設備端, 3, TRANS, SW),
      故障様相(地絡, 12, KIND),
      故障区間(故障区間不明, 3, AREA).
rule1(3, 17, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '64bdd', D),
      設備端(変圧器の2次側, 32, TRANS, SW),
      系統状況(同期設備は事故遮断, 2111, SW),
      故障様相(地絡, 12, KIND),
      故障区間(故障区間不明, 3, AREA).
rule1(1, 18, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '64bp', D),
      設備端(変圧器の1次側, 31, TRANS, SW),
      故障様相(不明, 14, KIND),
      故障区間(自区間変圧器, 13, TRANS, AREA).
rule1(1, 19, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '67g', 1),
      設備端(ラインの受電端, 11, LINE, SW),
      故障様相(地絡, 12, KIND),
      故障区間(ラインの受電端から下流に3区間, 211, LINE, AREA).
rule1(1, 20, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '67g', 1),
      設備端(ラインの送電端, 12, LINE, SW),
      系統状況(平行回線運用, 111, LINE, PAIR),
      故障様相(地絡, 12, KIND),
      故障区間(自区間ライン, 11, LINE, AREA).
rule1(1, 21, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '78g', -1),
      設備端(母線の設備端, 2, BUS, SW),
      故障様相('1相地絡', 121, KIND),
      故障区間(自区間母線, 12, BUS, AREA).
rule1(1, 22, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '78g', 1),
      設備端(ラインの設備端, 1, LINE, SW),
      故障様相('1相地絡', 121, KIND),
      故障区間(自区間ライン, 11, LINE, AREA).
rule1(1, 23, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '78s', -1),
      設備端(母線の設備端, 2, BUS, SW),
      故障様相(短絡, 11, KIND),
      故障区間(自区間母線, 12, BUS, AREA).
rule1(1, 24, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '78s', 1),
      設備端(母線の設備端, 2, LINE, SW),
      故障様相(短絡, 11, KIND),
      故障区間(自区間母線, 12, LINE, AREA).
rule1(1, 25, [RY], [RY, AREA, KIND]) :-
      系統状況(リレーが動作, 222, SW, RY, '85g', D),
      設備端(ラインの設備端, 1, LINE, SW),
      故障様相('1相地絡', 121, KIND),
```

```
        故障区間（自区間ライン，11，LINE，AREA）．
rule1（1，26，［RY］，［RY，AREA，KIND］）：－
        系統状況（リレーが動作，222，SW，RY，'85s'，D），
        設備端（ラインの設備端，1，LINE，SW），
        故障様相（短絡，11，KIND），
        故障区間（自区間ライン，11，LINE，AREA）．
rule1（1，27，［RY］，［RY，AREA，KIND］）：－
        系統状況（リレーが動作，222，SW，RY，'87'，D），
        設備端（変圧器の設備端，3，TRANS，SW），
        故障様相（不明，14，KIND），
        故障区間（自区間変圧器，13，TRANS，AREA）．
rule1（1，28，［RY］，［RY，AREA，KIND］）：－
        系統状況（リレーが動作，222，SW，RY，'87g'，－1），
        設備端（母線の設備端，2，BUS，SW），
        故障様相（'１相地絡'，121，KIND），
        故障区間（自区間母線，12，BUS，AREA）．
rule1（1，29，［RY］，［RY，AREA，KIND］）：－
        系統状況（リレーが動作，222，SW，RY，'87g'，1），
        設備端（ラインの設備端，1，LINE，SW），
        故障様相（'１相地絡'，121，KIND），
        故障区間（自区間ライン，11，LINE，AREA）．
rule1（1，30，［RY］，［RY，AREA，KIND］）：－
        系統状況（リレーが動作，222，SW，RY，'87s'，－1），
        設備端（母線の設備端，2，BUS，SW），
        故障様相（短絡，11，KIND），
        故障区間（自区間母線，12，BUS，AREA）．
rule1（1，31，［RY］，［RY，AREA，KIND］）：－
        系統状況（リレーが動作，222，SW，RY，'87s'，1），
        設備端（ラインの設備端，1，LINE，SW），
        故障様相（短絡，11，KIND），
        故障区間（自区間ライン，11，LINE，AREA）．
rule1（1，32，［RY］，［RY，AREA，KIND］）：－
        系統状況（リレーが動作，222，SW，RY，'95'，D），
        故障様相（不明，14，KIND），
        故障区間（故障区間不明，3，AREA）．
rule1（1，33，［RY］，［RY，AREA，KIND］）：－
        系統状況（リレーが動作，222，SW，RY，'96－2'，D），
        設備端（変圧器の設備端，3，TRANS，SW），
        故障様相（不明，14，KIND），
        故障区間（自区間変圧器，13，TRANS，AREA）．
rule1（1，34，［RY］，［RY，AREA，KIND］）：－
        系統状況（リレーが動作，222，SW，RY，'ホウアツカン'，D），
        設備端（変圧器の設備端，3，TRANS，SW），
        故障様相（不明，14，KIND），
        故障区間（自区間変圧器，13，TRANS，AREA）．
```

```
leps1916::>fd0>prim.esp

class    prim      has

 attribute
     atms;

:設備端(OBJ, DB, ラインの設備端, 1, LINE, SW) :-
      :db(DB, type, LINE, line, STAT),
      :db(DB, receive, LINE, LIST1),
      :db(DB, supply, LINE, LIST2),
      (:操作(OBJ, DB, メンバー, 2, SW, LIST1);
           :操作(OBJ, DB, メンバー, 2, SW, LIST2))
      ;

:設備端(OBJ, DB, ラインの受電端, 11, LINE, SW) :-
      :db(DB, type, LINE, line, STAT),
      :db(DB, receive, LINE, LIST1),
      :操作(OBJ, DB, メンバー, 2, SW, LIST1)
      ;

:設備端(OBJ, DB, ラインの送電端, 12, LINE, SW) :-
      :db(DB, type, LINE, line, STAT),
      :db(DB, supply, LINE, LIST2),
      :操作(OBJ, DB, メンバー, 2, SW, LIST2)
      ;

:設備端(OBJ, DB, 母線の設備端, 2, BUS, SW) :-
      :db(DB, type, BUS, bus, STAT),
      :db(DB, receive, BUS, LIST1),
      :db(DB, supply, BUS, LIST2),
      (:操作(OBJ, DB, メンバー, 2, SW, LIST1);
           :操作(OBJ, DB, メンバー, 2, SW, LIST2))
      ;

:設備端(OBJ, DB, 母線の母連, 21, BUS, SW) :-
      :db(DB, type, BUS, bus, STAT),
      (:db(DB, bus_tie, SW, BUS, BUS1);
       :db(DB, bus_tie, SW, BUS1, BUS))
      ;

:設備端(OBJ, DB, 母線の受電端, 22, BUS, SW) :-
      :db(DB, type, BUS, bus, STAT),
      :db(DB, receive, BUS, LIST1),
      :操作(OBJ, DB, メンバー, 2, SW, LIST1)
      ;

:設備端(OBJ, DB, 母線の送電端, 23, BUS, SW) :-
      :db(DB, type, BUS, bus, STAT),
      :db(DB, supply, BUS, LIST2),
      :操作(OBJ, DB, メンバー, 2, SW, LIST2)
      ;

:設備端(OBJ, DB, 変圧器の設備端, 3, TRANS, SW) :-
      :db(DB, type, TRANS, trans, STAT),
      :db(DB, receive, TRANS, LIST1),
      :db(DB, supply, TRANS, LIST2),
      (:操作(OBJ, DB, メンバー, 2, SW, LIST1);
           :操作(OBJ, DB, メンバー, 2, SW, LIST2))
      ;

:設備端(OBJ, DB, 変圧器の1次側, 31, TRANS, SW) :-
```

```
        :db(DB, type, TRANS, trans, STAT),
        :db(DB, receive, TRANS, LIST1),
        :操作(OBJ, DB, メンバー, 2, SW, LIST1)
        :

:設備端(OBJ, DB, 変圧器の2次側, 32, TRANS, SW):-
        :db(DB, type, TRANS, trans, STAT),
        :db(DB, supply, TRANS, LIST2),
        :操作(OBJ, DB, メンバー, 2, SW, LIST2)
        :

:故障様相(OBJ, DB, 故障, 1, KIND):-
        KIND=事故
        :
:故障様相(OBJ, DB, 短絡, 11, KIND):-
        KIND=短絡
        :
:故障様相(OBJ, DB, '2相短絡', 111, KIND):-
        KIND='2相短絡'
        :
:故障様相(OBJ, DB, '3相短絡', 112, KIND):-
        KIND='3相短絡'
        :
:故障様相(OBJ, DB, 地絡, 12, KIND):-
        KIND=地絡
        :
:故障様相(OBJ, DB, '1相地絡', 121, KIND):-
        KIND='1相地絡'
        :
:故障様相(OBJ, DB, '2相地絡', 122, KIND):-
        KIND='2相地絡'
        :
:故障様相(OBJ, DB, 断線, 13, KIND):-
        KIND=断線
        :
:故障様相(OBJ, DB, 断線短絡, 131, KIND):-
        KIND=断線短絡
        :
:故障様相(OBJ, DB, 断線地絡, 132, KIND):-
        KIND=断線地絡
        :
:故障様相(OBJ, DB, 不明, 14, KIND):-
        KIND=不明
        :
:故障様相(OBJ, DB, 短絡メンバー, 21, KIND):-
        member_of(KIND, [地絡, '1相地絡', '2相地絡'])
        :
:故障様相(OBJ, DB, 地絡メンバー, 22, KIND):-
        member_of(KIND, [短絡, '2相短絡', '3相短絡'])
        :

:故障区間(OBJ, DB, 自区間, 1, FACILITY, AREA):-
        :db(DB, type, FACILITY, _, _),
        AREA=[FACILITY]
        :
:故障区間(OBJ, DB, 自区間ライン, 11, LINE, AREA):-
        :db(DB, type, LINE, line, _),
        AREA=[LINE]
        :
:故障区間(OBJ, DB, 自区間母線, 12, BUS, AREA):-
        :db(DB, type, BUS, bus, _),
        AREA=[BUS]
        :
:故障区間(OBJ, DB, 自区間変圧器, 13, TRANS, AREA):-
        :db(DB, type, TRANS, trans, _),
        AREA=[TRANS]
```

```
        ;
:故障区間(OBJ, DB, '3区間', 2, FACILITY, AREA):-
      (  :db(DB, type, FACILITY, switch, _),
         :db(DB, supply, FACILITY, [FACILITY2]),
         :故障区間(OBJ, DB, '3区間', 2, FACILITY2, AREA)
      ;  :db(DB, type, FACILITY, line, _),
         :故障区間(OBJ, DB, ラインから3区間, 21, FACILITY, AREA)
      ;  :db(DB, type, FACILITY, bus, _),
         :故障区間(OBJ, DB, 母線から3区間, 22, FACILITY, AREA)
      )
        ;
:故障区間(OBJ, DB, ラインから3区間, 21, LINE, AREA):-
      :db(DB, type, LINE, line, _),
      :db(DB, supply, LINE, SW),
      supply1(DB, SW, LIST1, []),
      supply1(DB, LIST1, LIST2, []),
      supply1(DB, LIST2, LIST3, []),
      append([LINE], LIST1, BUF),
      append(BUF, LIST3, AREA1),
      dupchk(AREA1, AREA)
        ;
:故障区間(OBJ, DB, ラインの受電端から下流に3区間, 211, LINE, AREA):-
      :db(DB, type, LINE, line, _),
      :db(DB, supply, LINE, SW),
      supply1(DB, SW, LIST1, []),
      supply1(DB, LIST1, LIST2, []),
      supply1(DB, LIST2, LIST3, []),
      append([LINE], LIST1, BUF),
      append(BUF, LIST3, AREA1),
      dupchk(AREA1, AREA)
        ;
:故障区間(OBJ, DB, ラインの送電端から下流に3区間, 212, LINE, AREA):-
      :db(DB, type, LINE, line, _),
      :db(DB, supply, LINE, SW),
      supply1(DB, SW, LIST1, []),
      supply1(DB, LIST1, LIST2, []),
      supply1(DB, LIST2, LIST3, []),
      supply1(DB, LIST3, LIST4, []),
      supply1(DB, LIST4, LIST5, []),
      append(LIST1, LIST3, BUF),
      append(BUF, LIST5, AREA1),
      dupchk(AREA1, AREA)
        ;
:故障区間(OBJ, DB, ラインの受電端から上流に3区間, 213, LINE, AREA):-
      :db(DB, type, LINE, line, _),
      :db(DB, receive, LINE, SW),
      receive1(DB, SW, LIST1, []),
      receive1(DB, LIST1, LIST2, []),
      receive1(DB, LIST2, LIST3, []),
      receive1(DB, LIST3, LIST4, []),
      receive1(DB, LIST4, LIST5, []),
      append(LIST1, LIST3, BUF),
      append(BUF, LIST5, AREA1),
      dupchk(AREA1, AREA)
        ;
:故障区間(OBJ, DB, ラインの送電端から上流に3区間, 214, LINE, AREA):-
      :db(DB, type, LINE, line, _),
      :db(DB, receive, LINE, SW),
      receive1(DB, SW, LIST1, []),
      receive1(DB, LIST1, LIST2, []),
      receive1(DB, LIST2, LIST3, []),
      append([LINE], LIST1, BUF),
      append(BUF, LIST3, AREA1),
      dupchk(AREA1, AREA)
        ;
:故障区間(OBJ, DB, 母線から3区間, 22, BUS, AREA):-
```

```
        :db(DB, type, BUS, bus, _),
        :db(DB, supply, BUS, SW),
        supply1(DB, SW, LIST1, []),
        supply1(DB, LIST1, LIST2, []),
        supply1(DB, LIST2, LIST3, []),
        append([BUS], LIST1, BUF),
        append(BUF, LIST3, AREA1),
        dupchk(AREA1, AREA)
        ;
:故障区間(OBJ, DB, 母線の受電端から下流に3区間, 221, BUS, AREA):-
        :db(DB, type, BUS, bus, _),
        :db(DB, supply, BUS, SW),
        supply1(DB, SW, LIST1, []),
        supply1(DB, LIST1, LIST2, []),
        supply1(DB, LIST2, LIST3, []),
        append([BUS], LIST1, BUF),
        append(BUF, LIST3, AREA1),
        dupchk(AREA1, AREA)
        ;
:故障区間(OBJ, DB, 母線の送電端から下流に3区間, 222, BUS, AREA):-
        :db(DB, type, BUS, bus, _),
        :db(DB, supply, BUS, SW),
        supply1(DB, SW, LIST1, []),
        supply1(DB, LIST1, LIST2, []),
        supply1(DB, LIST2, LIST3, []),
        supply1(DB, LIST3, LIST4, []),
        supply1(DB, LIST4, LIST5, []),
        append(LIST1, LIST3, BUF),
        append(BUF, LIST5, AREA1),
        dupchk(AREA1, AREA)
        ;
:故障区間(OBJ, DB, 母線の受電端から上流に3区間, 223, BUS, AREA):-
        :db(DB, type, BUS, bus, _),
        :db(DB, receive, BUS, SW),
        receive1(DB, SW, LIST1, []),
        receive1(DB, LIST1, LIST2, []),
        receive1(DB, LIST2, LIST3, []),
        receive1(DB, LIST3, LIST4, []),
        receive1(DB, LIST4, LIST5, []),
        append(LIST1, LIST3, BUF),
        append(BUF, LIST5, AREA1),
        dupchk(AREA1, AREA)
        ;
:故障区間(OBJ, DB, 母線の送電端から上流に3区間, 224, BUS, AREA):-
        :db(DB, type, BUS, bus, _),
        :db(DB, receive, BUS, SW),
        receive1(DB, SW, LIST1, []),
        receive1(DB, LIST1, LIST2, []),
        receive1(DB, LIST2, LIST3, []),
        append([BUS], LIST1, BUF),
        append(BUF, LIST3, AREA1),
        dupchk(AREA1, AREA)
        ;
:故障区間(OBJ, DB, 故障区間不明, 3, AREA):-AREA=[]
        ;
:系統状況(OBJ, _, ルール1の採用結果, 43, RY, no, AREA, KIND):-
        :get(OBJ!atms, rule1, 1, 0, [RY], [_, AREA, KIND])
        ;
:系統状況(OBJ, DB, 平行回線運用, 111, LINE1, LINE2):-
        :db(DB, type, LINE1, line, off),
        :db(DB, type, LINE2, line, on),
        :db(DB, parallel, LINE1, LINE2)
        ;
:系統状況(OBJ, DB, 放射状運用, 112, LINE):-
        :db(DB, type, LINE, line, _),
        ( :db(DB, parallel, LINE, LINE2), !,
```

```
            ( :db(DB, receive, LINE, LIST1),
              offchk(DB, LIST1);
              :db(DB, receive, LINE2, LIST2),
              offchk(DB, LIST2)
            )
        ;true )
        .
:系統状況(OBJ, DB, 対向端の開閉設備, 1213, LINE, SW, SW1) :-
        :db(DB, receive, LINE, LIST),
        member_of(SW, LIST),
        :db(DB, supply, LINE, SW1)
        .
:系統状況(OBJ, DB, 対向端の開閉設備, 1213, LINE, SW, SW1) :-
        :db(DB, supply, LINE, LIST),
        member_of(SW, LIST),
        :db(DB, receive, LINE, SW1)
        .
:系統状況(OBJ, DB, 停電, 1221, LINE) :-
        :db(DB, type, LINE, line, off)
        .
:系統状況(OBJ, DB, 充電, 1222, LINE) :-
        :db(DB, type, LINE, line, on)
        .
:系統状況(OBJ, DB, 開閉設備は開, 211, SW) :-
        list(SW),
        offchk1(DB, SW)
        .
:系統状況(OBJ, DB, 開閉設備は開, 211, SW) :-
        :db(DB, sw, SW, _, off)
        .
:系統状況(OBJ, DB, 開閉設備は事故遮断, 2111, [SW]) :-
        :db(DB, sw, SW, on, off)
        .
:系統状況(OBJ, DB, 開閉設備は事故遮断, 2111, SW) :-
        :db(DB, sw, SW, on, off)
        .
:系統状況(OBJ, DB, 開閉設備は運用上開, 2112, SW) :-
        atomic(SW),
        :db(DB, sw, SW, off, off)
        .
:系統状況(OBJ, DB, 開閉設備は運用上開, 2112, SW) :-
        list(SW),
        offchk2(DB, SW)
        .
:系統状況(OBJ, DB, 開閉設備は閉, 212, SW) :-
        (atomic(SW),
         :db(DB, sw, SW, BEFORE, on)
        ;list(SW),
         onchk(DB, SW)
        ;(bound(SW), true
         ;  :db(DB, sw, SW, BEFORE, on)
         )
        )
        .
:系統状況(OBJ, DB, リレー設置, 221, SW, RY, DEVICE) :-
        :db(DB, relay, RY, DEVICE, SW, DIR, STAT)
        .
:系統状況(OBJ, DB, リレーが動作, 222, SW, RY, DEVICE, DIR) :-
        :db(DB, relay, RY, DEVICE, SW, DIR, on)
        .
:系統状況(OBJ, DB, テストケースの入出力, 3, [X], [Y, Z]) :-
        :db(DB, test, _, [X], [Y, Z])
        .
:系統状況(OBJ, DB, 全てオフ, 911, SETUBI) :-
        offcheck(DB, SETUBI)
        .
```

```
:系統状況(OBJ, DB, 全てオン, 912, SETUBI):-
      oncheck(DB, SETUBI)
      ;

:リレー種別(OBJ, DB, 短絡用リレー, 11, RY):-
      :db(DB, relay, RY, DEVICE, _, _, _),
      member_of(DEVICE, ['17s', '44s1', '44s2', '50s',
                         '51s', '51str', '78s', '85s',
                         '87s', '75'])
      ;
:リレー種別(OBJ, DB, 地絡用リレー, 12, RY):-
      :db(DB, relay, RY, DEVICE, _, _, _),
      member_of(DEVICE, ['50g', '17g', '85g', '67g',
                         '64b', '64bdd', '78g', '87g',
                         '67bp'])
      ;
:リレー種別(OBJ, DB, 主保護リレー, 21, RY):-
      :db(DB, relay, RY, DEVICE, _, _, _),
      member_of(DEVICE, ['17s', '44s1', '50s', '78s',
                         '87s', '50g', '17g', '78g',
                         '87g'])
      ;
:リレー種別(OBJ, DB, 後備保護リレー, 22, RY):-
      :db(DB, relay, RY, DEVICE, _, _, _),
      member_of(DEVICE, ['44s2', '51s', '51str', '67g'
                         '64b', '64bdd'])
      ;
:リレー種別(OBJ, DB, 無方向リレー, 23, RY):-
      :db(DB, relay, RY, DEVICE, _, _, _),
      member_of(DEVICE, ['27'])
      ;

:操作(OBJ, DB, 等しい, 11, X, Y):-
      string(X, _, _),
      equal_string(X, Y)
      ;
:操作(OBJ, DB, 等しい, 11, X, Y):-
      unbound(X),
      unbound(Y), cut_and_fail;
:操作(OBJ, DB, 等しい, 11, X, Y):-
      unbound(X),
      bound(Y), X=Y;
:操作(OBJ, DB, 等しい, 11, X, Y):-
      unbound(Y),
      bound(X), X=Y;
:操作(_, _, 等しい, 11, X, X);

:操作(_, _, 異なる, 12, X, Y):-
      string(X, _, _), string(Y, _, _), !,
      not_equal_string(X, Y)
      ;
:操作(_, _, 異なる, 12, X, Y):-X=\=Y
      ;
:操作(OBJ, DB, メンバー, 2, X, LIST):-
      member_of(X, LIST)
      ;
:操作(OBJ, DB, 要素が等しい, 31, L1, L2):-
      check_list(L1, L2),
      check_list(L2, L1)
      ;
:操作(OBJ, DB, 共通要素なし, 32, L1, L2):-
      member_one(L1, L2), !,
      member_one(L2, L1);

local
```

```
supply1(_, [], ANSWER, ANSWER) :- !;
supply1(DB, [X|REST], ANSWER, BUF) :-
          :db(DB, supply, X, LIST),
          append(BUF, LIST, NEW),
          !, supply1(DB, REST, ANSWER, NEW);
supply1(DB, [X|REST], ANSWER, BUF) :-
          !, supply1(DB, REST, ANSWER, BUF);

receive1(_, [], ANSWER, ANSWER) :- !;
receive1(DB, [X|REST], ANSWER, BUF) :-
          :db(DB, receive, X, LIST),
          append(BUF, LIST, NEW),
          !, receive1(DB, REST, ANSWER, NEW);
receive1(DB, [X|REST], ANSWER, BUF) :-
          !, receive1(DB, REST, ANSWER, BUF);

append([], LIST, LIST) :- !;
append([X|REST], LIST, ANSWER) :-
          append(REST, LIST, NEW),
          ANSWER=[X|NEW];

offchk(DB, [X|REST]) :-
          :db(DB, sw, X, off, AFTER);

offcheck(_, []) :- !;
offcheck(DB, [X|REST]) :-
          :db(DB, type, X, _, off), !,
          offcheck(DB, REST)
          ;

oncheck(_, []) :- !;
oncheck(DB, [X|REST]) :-
          :db(DB, type, X, _, on), !,
          oncheck(DB, REST)
          ;

onchk(_, []) :- !;
onchk(DB, [X|REST]) :-
          :db(DB, sw, X, BEFORE, on),
          onchk(DB, REST);

offchk1(_, []) :- !;
offchk1(DB, [X|REST]) :-
          :db(DB, sw, X, _, off),
          offchk1(DB, REST)
          ;

offchk2(_, []) :- !;
offchk2(DB, [X|REST]) :-
          :db(DB, sw, X, off, off),
          offchk2(DB, REST);

member_of(X, LIST) :-
          atomic(X),
          member(X, LIST);
member_of(X, LIST) :-
          string(X, _, _),
          member2(X, LIST);

member(X, [X|_]) :- !;
member(X, [_|REST]) :-
          member(X, REST);

member2(X, [Y|_]) :-equal_string(X, Y), !;
member2(X, [_|REST]) :-
          member2(X, REST);
```

```
dupchk([X|[]],[X]):-!;
dupchk([X|REST],ANSWER):-
            dupchk(REST,NEW),
            (member(X,NEW),!,
             ANSWER=NEW
            ;ANSWER=[X|NEW]
            );

check_list([],_);
check_list([X|REST],LIST):-
            member(X,LIST),!,
            check_list(REST,LIST)
            ;
member_one([],_):-!;
member_one([X|Rest],L2):-
            not(member(X,L2)),!,
            member_one(Rest,L2)
            ;
end.
```

```
tepsi916::>fd0>db.db

db(type,ab幹11,line,on).
db(type,ab幹21,line,on).
db(type,bc幹11,line,on).
db(type,bc幹21,line,on).
db(type,cd幹11,line,on).
db(type,cd幹21,line,on).
db(type,a母線,bus,on).
db(type,b母線,bus,on).
db(type,c母線,bus,on).
db(type,d母線,bus,on).
db(type,s0,switch,on).
db(type,s1,switch,on).
db(type,s2,switch,on).
db(type,s3,switch,on).
db(type,s4,switch,on).
db(type,s5,switch,on).
db(type,s6,switch,on).
db(type,s7,switch,on).
db(type,s8,switch,on).
db(type,s9,switch,on).
db(type,s10,switch,on).
db(type,s11,switch,on).
db(type,s12,switch,on).
db(receive,a母線,[s0]).
db(receive,b母線,[s2,s4]).
db(receive,c母線,[s6,s8]).
db(receive,d母線,[s10,s12]).
db(receive,ab幹11,[s1]).
db(receive,ab幹21,[s3]).
db(receive,bc幹11,[s5]).
db(receive,bc幹21,[s7]).
db(receive,cd幹11,[s9]).
db(receive,cd幹21,[s11]).
db(receive,s0,[]).
db(receive,s1,[a母線]).
db(receive,s2,[ab幹11]).
db(receive,s3,[a母線]).
db(receive,s4,[ab幹21]).
db(receive,s5,[b母線]).
db(receive,s6,[bc幹11]).
db(receive,s7,[b母線]).
db(receive,s8,[bc幹21]).
db(receive,s9,[c母線]).
db(receive,s10,[cd幹11]).
db(receive,s11,[c母線]).
db(receive,s12,[cd幹21]).
db(supply,a母線,[s1,s3]).
db(supply,b母線,[s5,s7]).
db(supply,c母線,[s9,s11]).
db(supply,d母線,[]).
db(supply,ab幹11,[s2]).
db(supply,ab幹21,[s4]).
db(supply,bc幹11,[s6]).
db(supply,bc幹21,[s8]).
db(supply,cd幹11,[s10]).
db(supply,cd幹21,[s12]).
db(supply,s0,[a母線]).
db(supply,s1,[ab幹11]).
db(supply,s2,[b母線]).
db(supply,s3,[ab幹21]).
```

```
db(supply, s4, [b母線]).
db(supply, s5, [bc線1]).
db(supply, s6, [c母線]).
db(supply, s7, [bc線2]).
db(supply, s8, [c母線]).
db(supply, s9, [cd線1]).
db(supply, s10, [d母線]).
db(supply, s11, [cd線2]).
db(supply, s12, [d母線]).
db(sw, s0, on, on).
db(sw, s1, on, on).
db(sw, s2, on, on).
db(sw, s3, on, on).
db(sw, s4, on, on).
db(sw, s5, on, on).
db(sw, s6, on, on).
db(sw, s7, on, on).
db(sw, s8, on, on).
db(sw, s9, on, on).
db(sw, s10, on, on).
db(sw, s11, on, on).
db(sw, s12, on, on).
db(relay, r10, '50s', s1, 1, off).
db(relay, r20, '50s', s2, 1, off).
db(relay, r30, '50s', s3, 1, off).
db(relay, r40, '50s', s4, 1, off).
db(relay, r50, '50s', s5, 1, off).
db(relay, r60, '50s', s6, 1, off).
db(relay, r70, '50s', s7, 1, off).
db(relay, r80, '50s', s8, 1, off).
db(relay, r90, '50s', s9, 1, off).
db(relay, r100, '50s', s10, 1, off).
db(relay, r110, '50s', s11, 1, off).
db(relay, r120, '50s', s12, 1, off).
db(relay, r11, '51s', s1, 1, off).
db(relay, r21, '51s', s2, 1, off).
db(relay, r31, '51s', s3, 1, off).
db(relay, r41, '51s', s4, 1, off).
db(relay, r51, '51s', s5, 1, off).
db(relay, r61, '51s', s6, 1, off).
db(relay, r71, '51s', s7, 1, off).
db(relay, r81, '51s', s8, 1, off).
db(relay, r91, '51s', s9, 1, off).
db(relay, r101, '51s', s10, 1, off).
db(relay, r111, '51s', s11, 1, off).
db(relay, r121, '51s', s12, 1, off).
db(relay, r64, '78s', s6, 1, off).
db(relay, r84, '78s', s8, -1, off).
db(parallel, ab線11, ab線21).
db(parallel, ab線21, ab線11).
db(parallel, bc線11, bc線21).
db(parallel, bc線21, bc線11).
db(parallel, cd線11, cd線21).
db(parallel, cd線21, cd線11).
```

```
teps1916::>fd0>tc4.tst

[db(test,1,[r24],{[b母線],短絡})],
[db(test,2,[r44],{[b母線],短絡})],
[db(test,3,[r50],{[bc線11],短絡})],
[db(test,4,[r64],{[bc線11],短絡})],
[db(relay,r24,'78s',s2,-1,off),
 db(relay,r24,'78s',s2,-1,on)],
[db(relay,r44,'78s',s4,-1,off),
 db(relay,r44,'78s',s4,-1,on)],
[db(type,b母線,bus,on),db(type,b母線,bus,off)],
[db(type,c母線,bus,on),db(type,c母線,bus,off)],
[db(type,bc線11,line,on),db(type,bc線11,line,off)],
[db(type,bc線21,line,on),db(type,bc線21,line,off)],
[db(type,cd線11,line,on),db(type,cd線11,line,off)],
[db(type,cd線21,line,on),db(type,cd線21,line,off)],
test(["RY"],[[r24],[r41]]),
[db(relay,r50,'50s',s5,1,off),
 db(relay,r50,'50s',s5,1,on)],
[db(relay,r64,'78s',s6,1,off),
 db(relay,r64,'78s',s6,1,on)],
[db(type,bc線11,line,on),db(type,bc線11,line,off)],
test(["RY"],[[r50],[r64]]).
```

```
tepsi916::>fd0>cr1.crule

error1(1, 2000, rule1, [RY, AREA, KIND], [2000]):-
        系統状況(テストケースの入出力, 3, [RY], [Y, Z]),
        操作(等しい, 11, KIND, Z),
        not(操作(要素が等しい, 31, AREA, Y)).

error1(1, 2001, rule1, [RY, AREA, KIND], [2001]):-
        系統状況(テストケースの入出力, 3, [RY], [Y, Z]),
        操作(要素が等しい, 31, AREA, Y),
        操作(異なる, 12, KIND, Z).

error1(1, 2002, rule1, [RY, AREA, KIND], [2002]):-
        not(系統状況(全てオフ, 911, AREA)).

error1(1, 2003, rule1, [RY, AREA, KIND], [2003]):-
        リレー種別(短絡用リレー, 11, RY),
        故障様相(短絡メンバー, 21, KIND).

error1(1, 2004, rule1, [RY, AREA, KIND], [2004]):-
        リレー種別(地絡用リレー, 12, RY),
        故障様相(地絡メンバー, 22, KIND).

error1(1, 2005, rule1, [RY, AREA, KIND], [2005]):-
        系統状況(テストケースの入出力, 3, [RY], [Y, Z]),
        操作(異なる, 12, KIND, Z).
```

```
tepsi916::>fd0>assum.inf

def(top_def,[power_system]).
def(power_system,[facility,protect_facility]).
def(facility,[side,state,connection]).
def(side,[line_side,bus_side,trans_side,switch_side]).
def(state,[line_state,bus_state,trans_state,switch_state]).
def(connection,[line_connection,bus_connection,
                trans_connection,switch_connection]).
def(line_state,[charge_l,parallel]).
def(line_side,[receive_l,supply_l]).
def(line_connection,[switch_l]).
def(bus_state,[charge_b,double_bus]).
def(bus_side,[bus_tie,receive_b,supply_b]).
def(bus_connection,[switch_b]).
def(trans_state,[charge_t,voltage_prime,voltage_second]).
def(trans_side,[prime,second,third]).
def(trans_connection,[switch_t]).
def(switch_state,[open_close,charge_s,member]).
def(switch_side,[receive_s,supply_s]).
def(switch_connection,[line_s,bus_s,trans_s]).
def(charge_l,[charge_on,charge_off]).
def(parallel,[para,non_para]).
def(open_close,[sw_open,sw_close]).
def(sw_open,[pre_fault,post_fault]).
def(protect_facility,[relay]).
def(relay,[relay_state,relay_class,protect_area,relay_kind]).
def(relay_state,[r_name,device,time,switch_name,action]).
def(relay_class,[principle_class,usage_class,kind_class]).
def(protect_area,['自区間','3区間','不明']).
def(relay_kind,[short,ground,dansen,unknown]).
def(device,['50s','51s','50g','17s','17g','44s1','44s2','44s3','11dd'
def(action,[action_on,action_off]).
def(principle_class,[line_member,bus_member,trans_member]).
def(usage_class,[main_member,back_up_member,non_member]).
def(kind_class,[short_member,ground_member]).
def('自区間',[ライン,母線,変圧器]).
def('3区間',[ラインから3区間,母線から3区間,変圧器から3区間]).
def('不明',[]).
def(short,['2LS','3LS']).
def(ground,['1LG','2LG']).
def(dansen,[断線距離,断線地絡]).
def(line_member,['50s','50g','44s1','44s2','51s','67g','64b','17s','1
def(bus_member,['87s','87g','44sbd','44sdd','64bc','64dd','51sbd']).
def(trans_member,['87','51','51s','64b']).
def(main_member,['50s','17s','44s1','50g']).
def(back_up_member,['44s2','44s3','44dd','51s','67g']).
def(non_member,['64b','27']).
def(ground_member,['50g','17g','67g','64b']).
def(short_member,['50s','44s1','44s2','51s','17s']).
def(ラインから3区間,[ラインの受電端から下流に3区間,
                      ラインの送電端から下流に3区間,
                      ラインの受電端から上流に3区間,
                      ラインの送電端から上流に3区間]).
def(母線から3区間,[母線から受電端から下流に3区間,
                    母線から送電端から下流に3区間,
                    母線から受電端から上流に3区間,
                    母線から送電端から上流に3区間]).
def(変圧器から3区間,['1次側から下流に3区間',
                      '1次側から上流に3区間',
                      '2次側から下流に3区間',
                      '2次側から上流に3区間']).
```

```
tepsl916::>fd0>flag.flg

flag(p_group,[設備端,設備端,[all]]).
flag(p_group,[故障様相,故障様相,[all]]).
flag(p_group,[故障区間,故障区間,[all]]).
flag(p_group,[系統状況,系統状況,[all]]).
flag(p_group,[リレー動作,系統状況,[222]]).
flag(p_group,[操作,操作,[all]]).
flag(p_group,[比較,操作,[11]]).
flag(p_group,[リレー種別,リレー種別,[all]]).
flag(strategy,[設備端,2,[]]).
flag(strategy,[故障様相,1,[]]).
flag(strategy,[故障区間,1,[]]).
flag(strategy,[系統状況,2,[]]).
flag(strategy,[リレー動作,2,[]]).
flag(strategy,[操作,2,[]]).
flag(ref,[line_side,[設備端(ラインの設備端,1,"LINE","SW")]]).
flag(ref,[bus_side,[設備端(母線の設備端,2,"BUS","SW")]]).
flag(ref,[trans_side,[設備端(変圧器の設備端,3,"TRANS","SW")]]).
flag(ref,[receive_l,[設備端(ラインの受電端,11,"LINE","SW")]]).
flag(ref,[supply_l,[設備端(ラインの送電端,12,"LINE","SW")]]).
flag(ref,[bus_tie,[設備端(母線の母連,21,"BUS","SW")]]).
flag(ref,[receive_b,[設備端(母線の受電端,22,"BUS","SW")]]).
flag(ref,[supply_b,[設備端(母線の送電端,23,"BUS","SW")]]).
flag(ref,[prime,[設備端(変圧器の1次側,31,"TRANS","SW")]]).
flag(ref,[second,[設備端(変圧器の2次側,32,"TRANS","SW")]]).
flag(ref,[nil,[故障様相(故障,1,"KIND")]]).
flag(ref,[short,[故障様相(短絡,11,"KIND")]]).
flag(ref,[ground,[故障様相(地絡,12,"KIND")]]).
flag(ref,[dansen,[故障様相(断線,13,"KIND")]]).
flag(ref,[unknown,[故障様相(不明,14,"KIND")]]).
flag(ref,['2LS',[故障様相('2相短絡',111,"KIND")]]).
flag(ref,['3LS',[故障様相('3相短絡',112,"KIND")]]).
flag(ref,['1LG',[故障様相('1相地絡',121,"KIND")]]).
flag(ref,['2LG',[故障様相('2相地絡',122,"KIND")]]).
flag(ref,[断線短絡,[故障様相(断線短絡,131,"KIND")]]).
flag(ref,[断線地絡,[故障様相(断線地絡,132,"KIND")]]).
flag(ref,[自区間,[故障区間(自区間,1,"FACILITY","AREA")]]).
flag(ref,['3区間',[故障区間('3区間',2,"FACILITY","AREA")]]).
flag(ref,[不明,[故障区間(故障区間不明,3,"AREA")]]).
flag(ref,[ライン,[故障区間(自区間ライン,11,"LINE","AREA")]]).
flag(ref,[母線,[故障区間(自区間母線,12,"BUS","AREA")]]).
flag(ref,[変圧器,[故障区間(自区間変圧器,13,"TRANS","AREA")]]).
flag(ref,[調相設備,[故障区間(自区間調相設備,14,"FACIL","AREA")]]).
flag(ref,[ラインから3区間,[故障区間(ラインから3区間,21,"LINE","AREA")]]).
flag(ref,[母線から3区間,[故障区間(母線から3区間,22,"LINE","AREA")]]).
flag(ref,[ラインの受電端から下流に3区間,[故障区間(ラインの受電端から下流に3区間,211,"LINE","AREA")]]).
flag(ref,[ラインの送電端から下流に3区間,[故障区間(ラインの送電端から下流に3区間,212,"LINE","AREA")]]).
flag(ref,[ラインの受電端から上流に3区間,[故障区間(ラインの受電端から上流に3区間,213,"LINE","AREA")]]).
flag(ref,[ラインの送電端から上流に3区間,[故障区間(ラインの送電端から上流に3区間,214,"LINE","AREA")]]).
flag(ref,[母線の受電端から下流に3区間,[故障区間(母線の受電端から下流に3区間,221,"BUS","AREA")]]).
flag(ref,[母線の送電端から下流に3区間,[故障区間(母線の送電端から下流に3区間,222,"BUS","AREA")]]).
flag(ref,[母線の受電端から上流に3区間,[故障区間(母線の受電端から上流に3区間,223,"BUS","AREA")]]).
flag(ref,[母線の送電端から上流に3区間,[故障区間(母線の送電端から上流に3区間,224,"BUS","AREA")]]).
flag(ref,[para,[系統状況(平行回線運用,111,"LINE1","LINE2")]]).
flag(ref,[non_para,[系統状況(放射状運用,112,"LINE")]]).
flag(ref,[distinguish,[系統状況(対向端の開閉設備,1213,"LINE","SW","SW1")]]).
flag(ref,[charge_off,[系統状況(停電,1221,"LINE")]]).
flag(ref,[charge_on,[系統状況(充電,1222,"LINE")]]).
flag(ref,[sw_open,[系統状況(開閉設備は開,211,"SW")]]).
flag(ref,[post_fault,[系統状況(開閉設備は事故遮断,2111,"SW1")]]).
flag(ref,[pre_fault,[系統状況(開閉設備は運用上開,2112,"SW1")]]).
```

```
flag(ref, [close, [系統状況(開閉設備識別, 212, "SW")]]).
flag(ref, [relay, [系統状況(リレー設置, 221, "SW", "RY", "DEVICE")]]).
flag(ref, [action_on, [系統状況(リレーが動作, 222, "SW", "RY", "DEVICE", "DIR")]]).
flag(ref, [nil, [系統状況(テストケースの入出力, 3, "IN", "OUT")]]).
flag(ref, [nil, [操作(等しい, 11, "X", "LIST")]]).
flag(ref, [nil, [操作(異なる, 12, "X", "LIST")]]).
flag(ref, [nil, [操作(メンバー, 2, "L1", "L2")]]).
flag(ref, [nil, [操作(要素が等しい, 31, "L1", "L2")]]).
flag(ref, [nil, [操作(共通要素なし, 32, "L1", "L2")]]).
flag(ref, [shout_member, [リレー種別(短絡用リレー, 11, "RY")]]).
flag(ref, [ground_member, [リレー種別(地絡用リレー, 12, "RY")]]).
flag(ref, [main_member, [リレー種別(主保護リレー, 21, "RY")]]).
flag(ref, [back_up_member, [リレー種別(後備保護リレー, 22, "RY")]]).
flag(ref, [non_member, [リレー種別(無方向リレー, 23, "RY")]]).
flag(error, [2000, [設備端, 系統状況, 故障区間]]).
flag(error, [2001, [設備端, 系統状況, 故障区間]]).
flag(error, [2002, [設備端, 系統状況, 故障区間]]).
flag(error, [2003, [設備端, 系統状況, 故障区間]]).
flag(error, [2004, [設備端, 系統状況, 故障様相]]).
flag(error, [2008, [操作, リレー種別, 系統状況]]).
```

A.3　検証結果

検証過程と検証結果に関する以下の表示画面を次ページ以降に示す。

1)対話ウィンド
　検証の開始から終了までの対話画面
　生成仮説と成功仮説（生成仮説の検証結果）の表示

2)推論ネットワーク
　ＡＴＭＳノードの表示

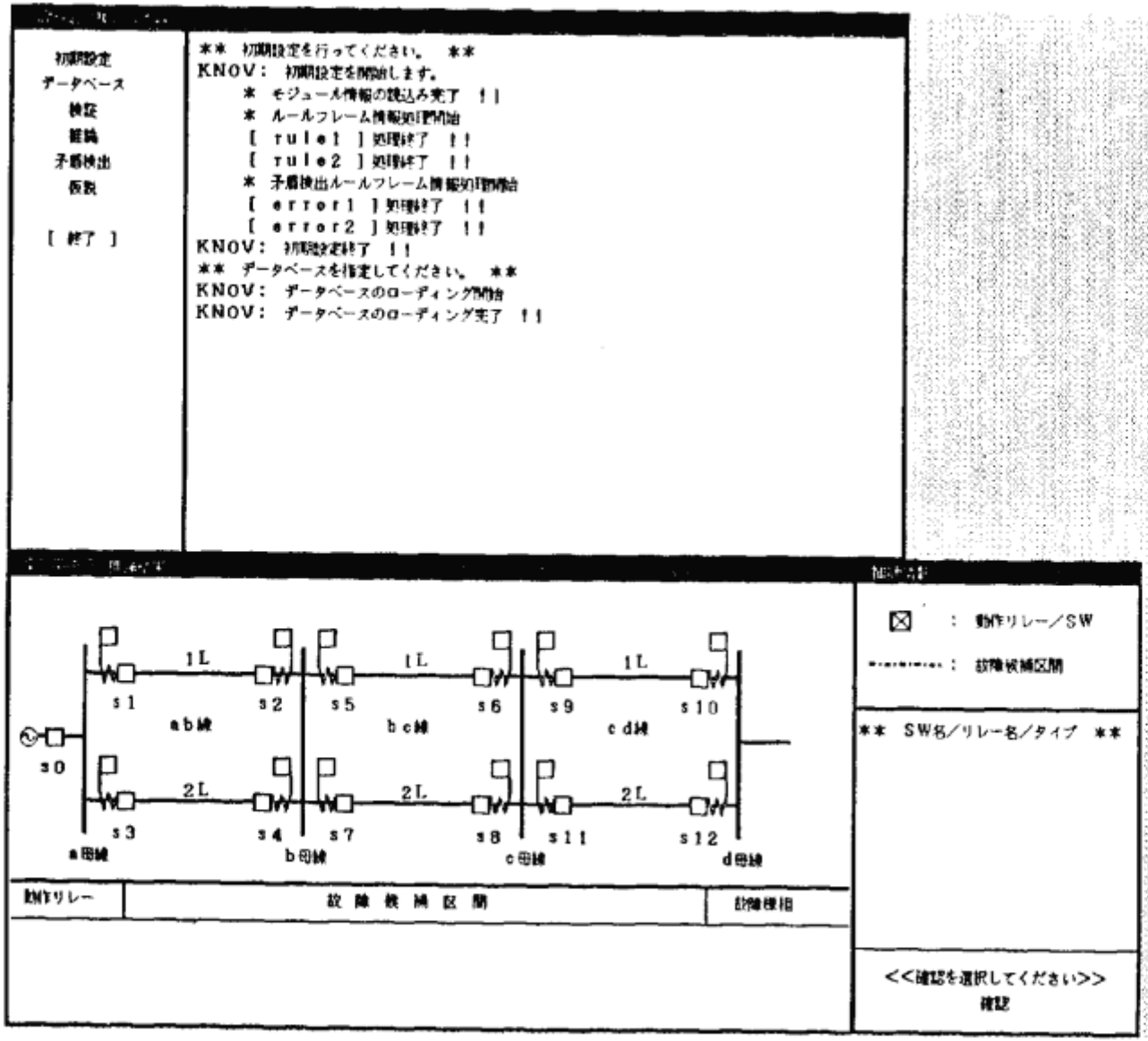
3)グラフィック表示例
　補助情報表示クラスとして"eg"を指定した場合の表示。
　この場合にはクラス"eg"を予めカタログしておく必要がある。

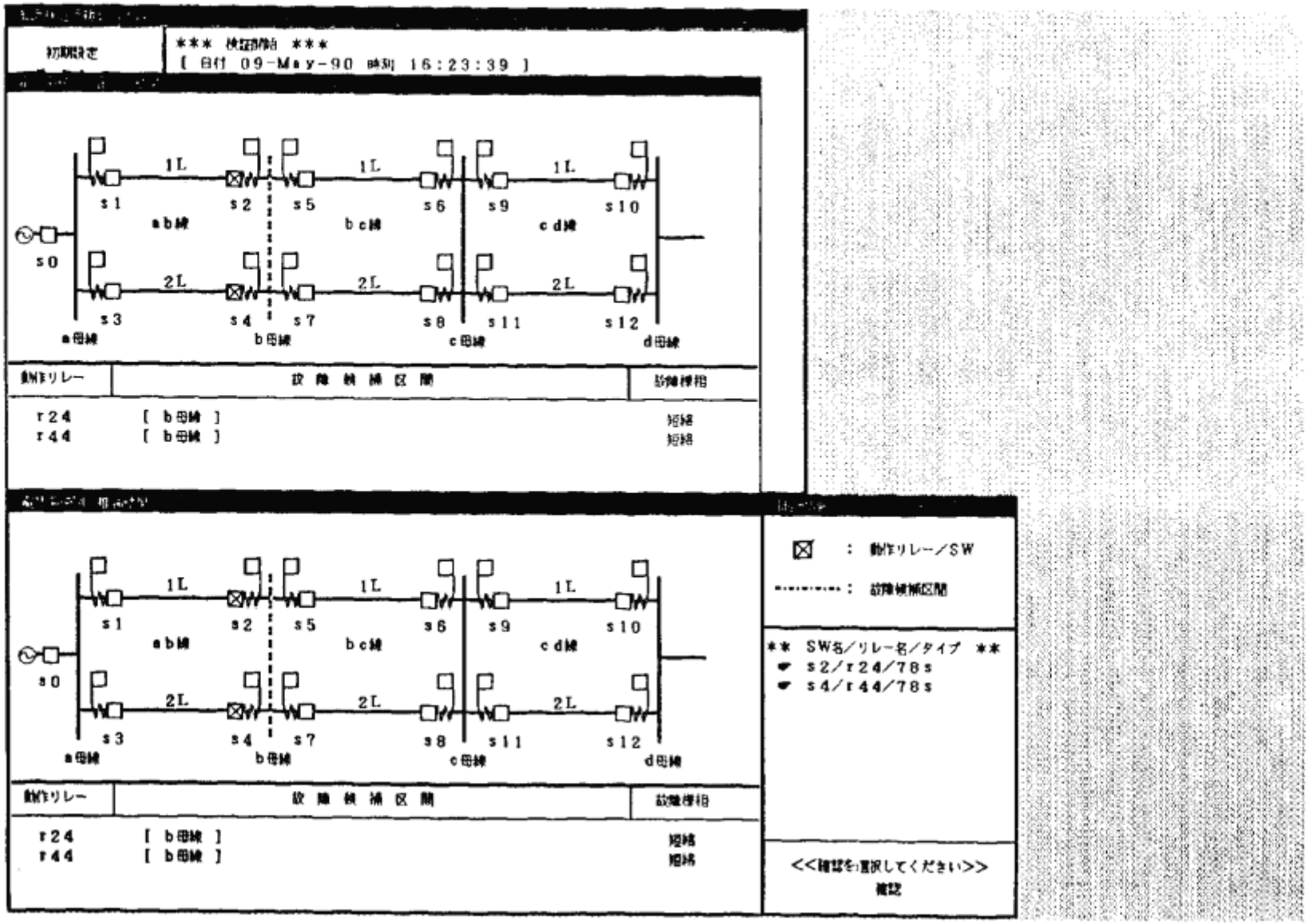
```
＊＊＊　検証開始　＊＊＊
[　時刻　16：27：21　]
　テストケース：　tc4.lst
　推論方式：　全探索　　>トレース：　nff
------------------------------------------------------------
＊＊＊　<　rule1　>
| 入力値：[r24]
-＞優先度=1　ID=23　succeed.
　　変数名：RY1　値：r24
　　変数名：AREA　値：[b母線]
　　変数名：KIND　値：短絡
＊＊＊　<　終了　>　＊＊＊
＊＊＊　<　rule1　>
| 入力値：[r44]
-＞優先度=1　ID=23　succeed.
　　変数名：RY1　値：r44
　　変数名：AREA　値：[b母線]
　　変数名：KIND　値：短絡
＊＊＊　<　終了　>　＊＊＊
∞∞∞　[　error1　]　矛盾検出ルール実行
∞∞∞　[　終了　]　∞∞∞
＊＊＊　<　rule1　>
| 入力値：[r50]
-＞優先度=1　ID=8　succeed.
　　変数名：RY1　値：r50
　　変数名：AREA　値：[bc線11]
　　変数名：KIND　値：短絡
＊＊＊　<　終了　>　＊＊＊
＊＊＊　<　rule1　>
| 入力値：[r64]
-＞優先度=1　ID=24　succeed.
　　変数名：RY1　値：r64
　　変数名：AREA　値：[c母線]
　　変数名：KIND　値：短絡
＊＊＊　<　終了　>　＊＊＊
∞∞∞　[　error1　]　矛盾検出ルール実行
∞∞∞　[　終了　]　∞∞∞
＊＊＊　<　rule2　>
| 入力値：[r50，r64，[c母線]，短絡]
-＞優先度=1　ID=1　succeed.
　　変数名：MAL_FUNC　値：r64
＊＊＊　<　終了　>　＊＊＊
＊＊＊　<　rule2　>
| 入力値：[r64，r50，[bc線11]，短絡]
-＞優先度=1　ID=1　succeed.
　　変数名：MAL_FUNC　値：r50
＊＊＊　<　終了　>　＊＊＊
∞∞∞　[　error2　]　矛盾検出ルール実行
∞∞∞　[　終了　]　∞∞∞
＊＊＊　矛盾検出により、次のルールが矛盾です。
　rule1(1，24，[A]，[A，B，C])：-
　　系統状況(リレーが動作，222，D，A，78s，1)，
　　設備項(母線の設備項，2，E，D)，
　　故障様相(短絡，11，C)，
　　故障区間(各区間母線，12，E，B)．
仮説生成を行いますか　(y，／n，)　?-y．

＊＊＊　仮説生成終了　[1]　＊＊＊

------------------------------------------------------------
[　時刻　16：27：55　]
```

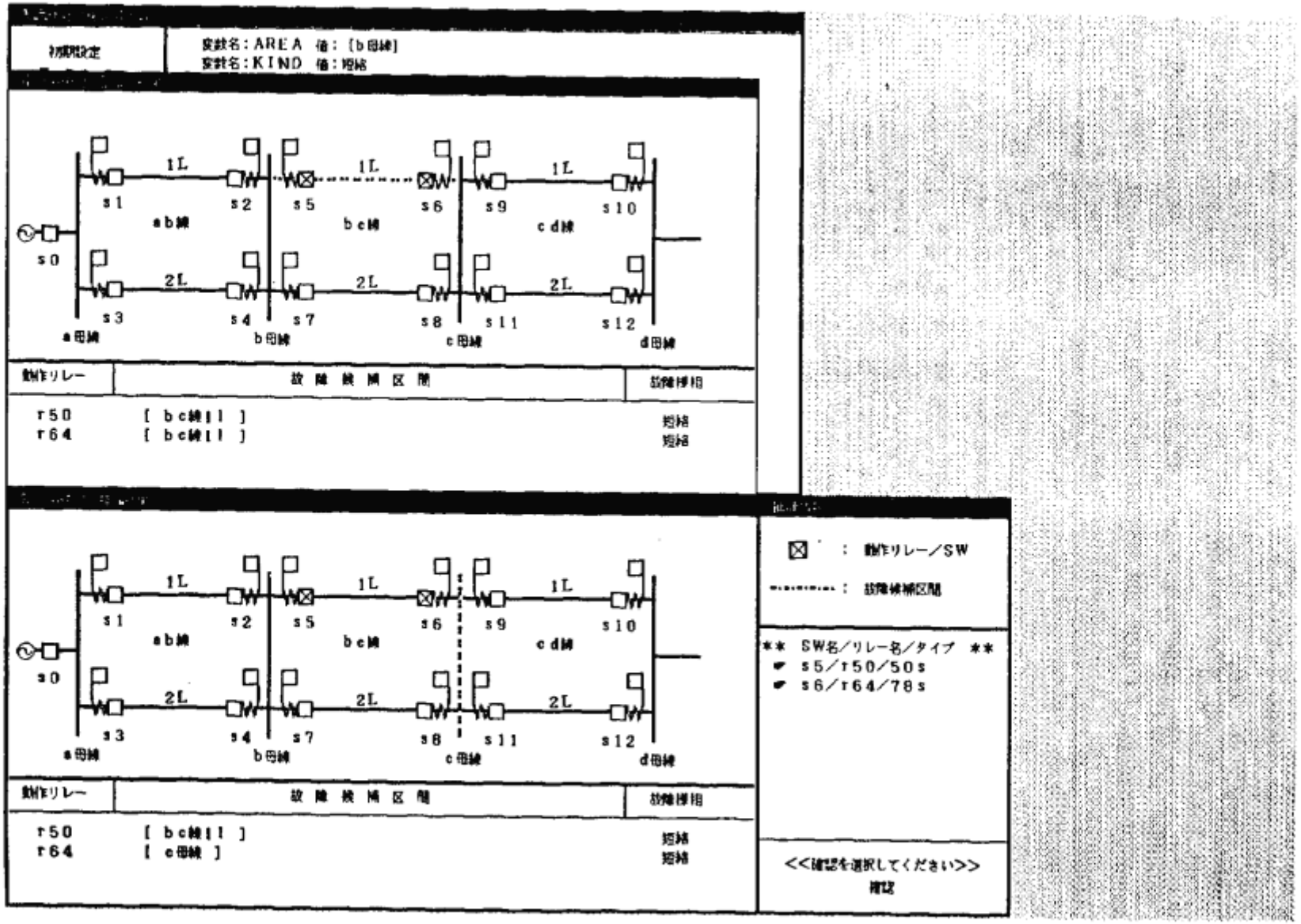
```
＊＊＊　検証終了　＊＊＊

＊＊＊　仮説表示開始　＊＊＊
□ルール名：ｒｕｌｅ１　　開始ＩＤ：　３５
□全仮説数－１０１　　□無予盾仮説数－５９　　□予盾仮説数　４２
□未成功仮説数－５８　　□成功仮説数－１　　□成功仮説数（同一結果）－０

ＩＤ表示　（ｉｄ．）　仮説表示　（ａｄ．）　終了　（ｅ．）
選択＞ｉｄ．
１．無予盾仮説　２．予盾仮説　３．未成功仮説　４．成功仮説　５．成功仮説（同一結果）
番号＞１．
【　成功仮説　】
　◆　１０５
番号＞ｅ．
ＩＤ表示　（ｉｄ．）　仮説表示　（ａｄ．）　終了　（ｅ．）
選択＞ａｄ．
表示する仮説ＩＤを入力してください．　終了（ｅ．）
ＩＤ＞１０５．
□ＩＤ－１０５　　仮説　　状態：無予盾
　ｒｕｌｅ１（１，１０５，［Ａ］，［Ａ，Ｂ，Ｃ］）：－
　　系統状況（リレー動作，２２２，Ｄ，Ａ，７８ｓ，１），
　　設備種（ラインの設備種，Ｉ，Ｅ，Ｄ），
　　故障箇所（短絡，１１，Ｃ），
　　故障区間（自区間ライン，１１，Ｅ，Ｂ）．
```

```
tepsi916::>fd0>tc43.log

***  インスタンス・ルールノード  表示開始  ***
☒ルール名：rule1  引数：1  ノード数：21    ○ 無矛盾    ⊥ 矛盾
○  node#1  導出ルールID：[23]
        入力値：[r24]
        出力値：[r24, [b母線], 短絡]
○  node#2  導出ルールID：[23]
        入力値：[r44]
        出力値：[r44, [b母線], 短絡]
○  node#3  導出ルールID：[8]
        入力値：[r50]
        出力値：[r50, [bc線11], 短絡]
⊥  node#4  導出ルールID：[24, 72]
        入力値：[r64]
        出力値：[r64, [c母線], 短絡]
⊥  node#5  導出ルールID：[35, 38, 68, 71, 102]
        入力値：[r64]
        出力値：[r64, [ab線11], 短絡]
⊥  node#6  導出ルールID：[36, 42, 69, 76, 103, 115]
        入力値：[r64]
        出力値：[r64, [a母線, ab線11, ab線21, b母線], 短絡]
⊥  node#7  導出ルールID：[37, 70, 104]
        入力値：[r64]
        出力値：[r64, [], 短絡]
⊥  node#8  導出ルールID：[41, 43, 75, 77]
        入力値：[r64]
        出力値：[r64, [ab線11, b母線, bc線11, bc線21], 短絡]
⊥  node#9  導出ルールID：[44, 78]
        入力値：[r64]
        出力値：[r64, [b母線, bc線11, bc線21, c母線], 短絡]
⊥  node#10  導出ルールID：[45, 79, 106, 118]
        入力値：[r64]
        出力値：[r64, [a母線], 短絡]
⊥  node#11  導出ルールID：[46, 80]
        入力値：[r64]
        出力値：[r64, [ab線11, a母線], 短絡]
⊥  node#12  導出ルールID：[47, 81, 112]
        入力値：[r64]
        出力値：[r64, [c母線, cd線11, cd線21, d母線], 短絡]
⊥  node#13  導出ルールID：[48, 82]
        入力値：[r64]
        出力値：[r64, [cd線11, cd線21, d母線], 短絡]
⊥  node#14  導出ルールID：[49, 83]
        入力値：[r64]
        出力値：[r64, [bc線11, bc線21, b母線, ab線11, ab線21], 短絡]
⊥  node#15  導出ルールID：[50, 84]
        入力値：[r64]
        出力値：[r64, [c母線, bc線11, bc線21, b母線], 短絡]
○  node#16  導出ルールID：[105]
        入力値：[r64]
        出力値：[r64, [bc線11], 短絡]
⊥  node#17  導出ルールID：[109, 111]
        入力値：[r64]
        出力値：[r64, [bc線11, c母線, cd線11, cd線21], 短絡]
⊥  node#18  導出ルールID：[113]
        入力値：[r64]
        出力値：[r64, [b母線, ab線11, ab線21, a母線], 短絡]
⊥  node#19  導出ルールID：[114]
        入力値：[r64]
        出力値：[r64, [bc線11, b母線, ab線11, ab線21], 短絡]
⊥  node#20  導出ルールID：[116]
```

```
      入力値：[r64]
      出力値：[r64，[ab線11，ab線21，b母線，bc線11，bc線21]，短絡]
上 node#2[  導出ルール[D：[117]
      入力値：[r64]
      出力値：[r64，[cd線11，cd線21，c母線，bc線11，bc線21]，短絡]
***  表示終了  ***
```

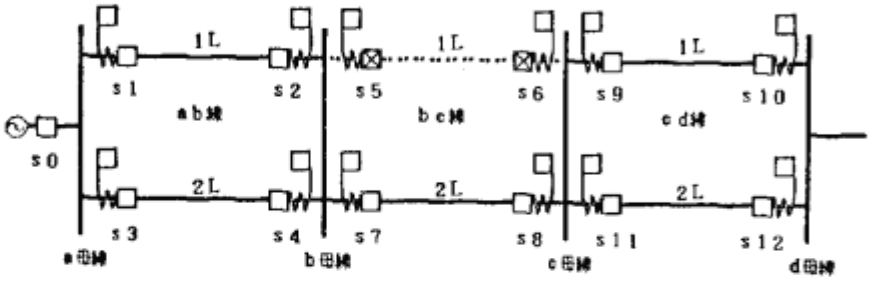

初期設定
データベース
検証
推論
予盾検出
仮説
[終了]
** 初期設定を行ってください。 **
KNOV: 初期設定を開始します。
* モジュール情報の読込み完了 !!
* ルールフレーム情報処理開始
[rule1] 処理終了 !!
[rule2] 処理終了 !!
* 予盾検出ルールフレーム情報処理開始
[error1] 処理終了 !!
[error2] 処理終了 !!
KNOV: 初期設定終了 !!
** データベースを指定してください。 **
KNOV: データベースのローディング開始
KNOV: データベースのローディング完了 !!
1L
2L
s0
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
s11
s12
a母線
b母線
c母線
d母線
ab線
bc線
cd線
動作リレー
故障候補区間
故障種別
: 動作リレー／SW
: 故障候補区間
** SW名／リレー名／タイプ **
<<確認を選択してください>>
確認


初期設定
*** 検証開始 ***
[日付 09-May-90 時刻 16:23:39]
1L
2L
s0
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
s11
s12
a b 線
b c 線
c d 線
a 母線
b 母線
c 母線
d 母線
動作リレー
故 障 検 補 区 間
故障様相
r24 [b 母線] 短絡
r44 [b 母線] 短絡
: 動作リレー／SW
: 故障検補区間
** SW名／リレー名／タイプ **
s2／r24／78s
s4／r44／78s
<<確認を選択してください>>
確認

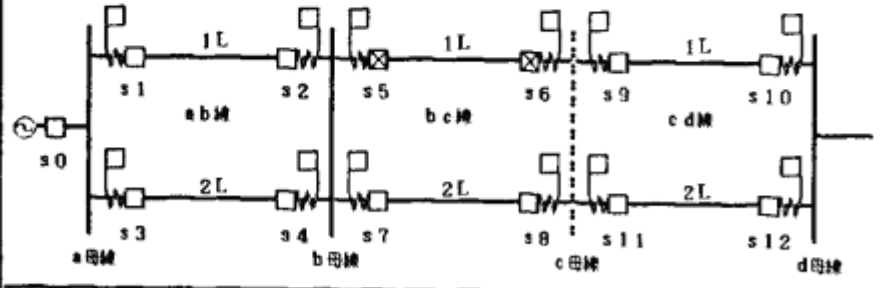

初期設定
変数名：AREA 値：[b母線]
変数名：KIND 値：短絡
1L
2L
s0
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
s11
s12
ab線
bc線
cd線
a母線
b母線
c母線
d母線
動作リレー
故障検出区間
故障種別
r50 [bc線1] 短絡
r64 [bc線1] 短絡
r50 [bc線1] 短絡
r64 [c母線] 短絡
：動作リレー／SW
：故障候補区間
** SW名／リレー名／タイプ **
s5／r50／50s
s6／r64／78s
<<確認を選択してください>>
確認

初期設定

ooooo [終了] ooooo
*** < rule2 >



動作リレー	故障検索区間	故障様相
r50	[bc線11]	短絡
r64	[bc線11]	短絡



動作リレー	故障検索区間	故障様相
r50	[bc線11]	短絡
r64	[c母線]	短絡

⊠ : 動作リレー／SW

----- : 故障検索区間

```
error1(1, 2000, rule1, [A, B, C], [2000]) :-
    系統状況(テストケースの入出力, 3, [A], [D, E]),
    操作(等しい, 11, C, E),
    not(操作(要素が等しい, 31, B, D)).
```

```
rule1(1, 24, [A], [A, B, C]) :-
    系統状況(リレーが動作, 222, D, A, '78s', 1),
    設備端(母線の設備端, 2, E, D),
    故障様相(3相短, 11, C),
    故障区間(自区間母線, 12, E, B).
```

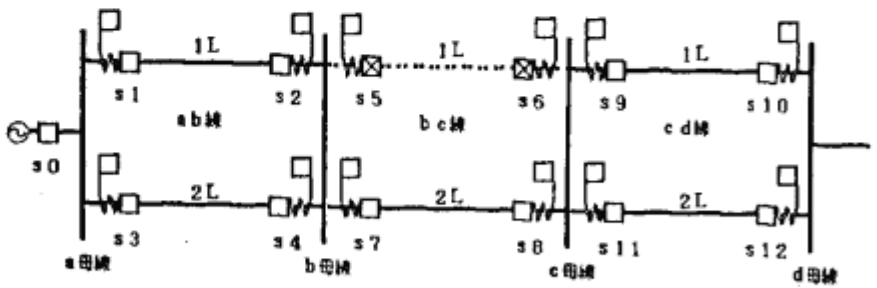
初期設定
データベース
検証
推論
予兆検出
仮説

[終了]

変数名：MAL_FUNC 値：r50
*** < 終了 > ***
ooooo [error2] 予兆検出ルール実行
ooooo [終了] ooooo
*** 予兆検出により、次のルールが予兆です。
rule1(1, 24, [A], [A, B, C]) :-
系統状況(リレーが動作, 222, D, A, '78s', 1),
設備端(母線の設備端, 2, E, D),
故障様相(短絡, 11, C),
故障区間(自区間母線, 12, E, B).
仮説生成を行いますか (y, /n.) ?-y.

*** 仮説生成終了 [1] ***

[時刻 16:29:05]
*** 検証終了 ***



動作リレー	故障検知区間	故障様相
r50	[bc線11]	短絡
r64	[bc線11]	短絡

rule1(1, 105, [A], [A, B, C]) :-
系統状況(リレーが動作, 222, D, A, '78s', 1),
設備端(ラインの設備端, 1, E, D),
故障様相(短絡, 11, C),
故障区間(自区間ライン, 11, E, B).

error1(1, 2000, rule1, [A, B, C], [2000]) :-
系統状況(テストケースの入出力, 3, [A], [D, E]),
操作(等しい, 11, C, E),
not(操作(要素が等しい, 31, B, D)).

rule1(1, 24, [A], [A, B, C]) :-
系統状況(リレーが動作, 222, D, A, '78s', 1),
設備端(母線の設備端, 2, E, D),
故障様相(短絡, 11, C),
故障区間(自区間母線, 12, E, B).

付録-B　計算機システム運用支援への適用

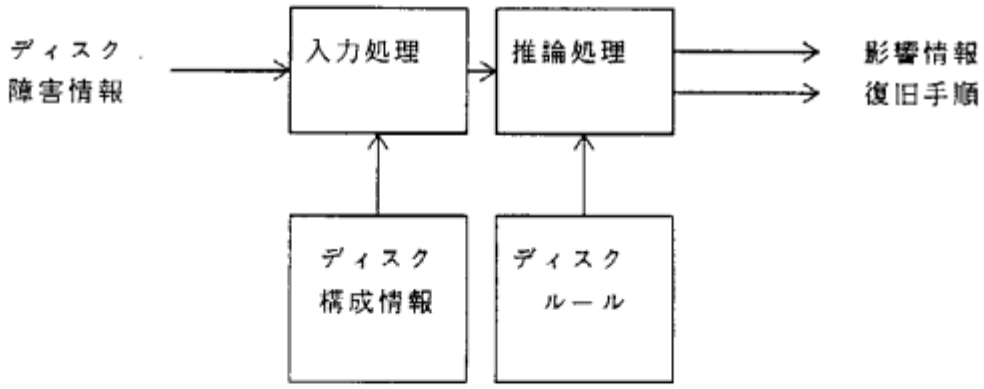
B.1　解説

B.1.1　計算機システム運用支援

計算機システムでは、ハードウェア障害が発生したときに、ＯＳによる自動復旧や、フィールドエンジニアによる障害機器同定、復旧処理の他に、システムを運用するオペレータ側で行わなければならない復旧処理がある。

たとえば、システム全体、あるいは、特定のディスク装置がダウンしたとき、磁気テープ、あるいは、ディスクのジャーナルから最新データを復元する。そのための手段は、システムが提供してくれるが、具体的に、どのバックアップからどういう手順で復元するか、復元してシステムを再起動するまでにどれだけ時間を要するか、業務を実行できない時間がどれ位長ければ、どのユーザに通知する必要があるかなど運用側でしか判断できない事柄がある。

計算機システム運用支援エキスパートシステムはこのような、システムの運用状況、構成情報、障害状況に応じた影響を判断し、復旧手順のガイダンスを出力する。以下に示す知識検証例ではそれらのなかでディスク障害の影響判定と復旧手順出力に範囲を絞っている。即ち、「障害ディスク」及び「障害ファイル」を与えると、推論の結果「障害の影響範囲、程度、復旧処置・手順」を得るエキスパートシステムを対象としている。本エキスパートシステムの構成を図B-1に示す。



図B-1　計算機システム（ディスク障害）復旧支援エキスパートシステムの構成

B.1.2　診断知識

診断ルール（フレーム）

(1)ディスクルール(rule1)

障害ディスクと、そのディスクの用途から障害の影響と処置・対策を判断する。

```
frame(rule1,['DISK'],['DISK','STEP','INF','DSU'],....,
            [[ディスク用途],[処理影響,ＴＳＳ影響,ディスク種類]])
```

DISK：障害ディスク
STEP：処置・状況
INF　：ＴＳＳへの影響
DSU　：ディスク用途

(2)ファイルルール(rule2)

障害ディスクに含まれるファイルと、そのファイルの運用情報、他のバックアップ（対）ディスクの状態などから障害の影響範囲と処置・対策を判断する。

```
frame(rule2,['FNAME','DISK'],
      ['FNAME','DISK','INF','SYSINF','STEP','FREASON','VSOP'],....,
        [[ファイル種類,ファイル特性,業務,ファイル包含,対ディスク状態],
        [ＴＳＳ影響,システム影響,処理影響,理由説明,ファイル影響グループ]])
```

FNAME　：障害ファイル
DISK　　：障害ディスク
INF　　：ＴＳＳ業務への影響
SYSINF　：システムへの影響
STEP　　：処置・状況
FREASON：理由説明
VSOP　　：ファイル影響度

(3)手順ルール(rule3)

rule1、rule2の出力である処置を入力として、具体的な操作手順に展開する。

```
frame(rule3,['STEP','WHDAY'],['STEP','WHDAY','DOP'],....,
                                        [[処理影響,日時],[手順展開]])
```

STEP　：処置・状況
WHDAY：障害日時
DOP　：復旧手順展開

基本概念（プリミティブグループ）

```
flag(p_group,[ディスク用途,ディスク用途,[all]]).
flag(p_group,[日時,日時,[all]]).
flag(p_group,[ＴＳＳ影響,ＴＳＳ影響,[all]]).
flag(p_group,[処理影響,処理影響,[all]]).
flag(p_group,[手順展開,手順展開,[all]]).
flag(p_group,[ディスク種類,ディスク種類,[all]]).
flag(p_group,[ファイル種類,ファイル種類,[all]]).
flag(p_group,[業務,業務,[all]]).
flag(p_group,[ファイル特性,ファイル特性,[all]]).
flag(p_group,[対ディスク状態,対ディスク状態,[all]]).
flag(p_group,[ファイル包含,ファイル包含,[all]]).
flag(p_group,[システム影響,システム影響,[all]]).
flag(p_group,[理由説明,理由説明,[all]]).
flag(p_group,[ファイル影響グループ,ファイル影響グループ,[all]]).
```

データベース（ディスク構成定義）

本エキスパートシステムではディスク故障を対象とするためデータベース定義としてはディスク構成について定義する。ディスク構成は述語'db'で定義する。第１引数が、定義内容のキーワードになっており、それぞれの定義内容に応じて引数を設定する。

表B-1　ディスク構成　述語定義

述語フォーマット	意味
db(disk,DISK,K,LIVE)	DISK　ディスク装置名 K　ディスク用途 LIVE　ディスクが正常か障害か(live,die)
db(file,FNAME,DISK,TASK,SORT,SUB)	FNAME　ファイル名 DISK　ディスク装置名 TASK　使用業務名 SORT　ファイル種類 SUB　補助情報
db(pair,FNAME,[DISK])	FNAME　ファイル名 DISK　ディスク装置名 （空は非多重化ファイル、リストは多重化ファイルの対ディスク）

B.1.3　矛盾検出知識

矛盾検出知識は以下に示す３種類に大別される。

1)テストケースとの矛盾
テストケースで設定している結果と、実際に推論を実行した結果とが異なることを言い、対象知識ベースの知識フレームには依存するが、知識そのものとは独立に定義できる。

2)ルール相互の論理的矛盾
知識の記述形式から検出できる矛盾である。たとえば、if（A）then（B）とif（A）then（not B）の両方が推論知識に含まれるような場合である。これも、知識そのものとは独立に定義できる。

3)その他の矛盾
対象の知識により異なり専門家またはＫＥが個別に定義するものである。プラグマティックな浅い知識の集積であり以下の様な断片的知識で矛盾を定義する。
・「ディスクの障害がシステムの運用に影響しないように
最低限の配慮はされているはずである」
・「ある特性を持つファイルの障害に対して、
その特性に関する推論結果が得られるはずである」

以下に示す矛盾検出知識は、上述の1)と3)を含む。

(1)rule1に関する矛盾ルール
・ディスクの用途が、テストケースの設定と異なる　（識別番号=2000）

(2)rule2に関する矛盾ルール
・推論結果が、テストケースの設定と異なる
処置・状況が異なる　（識別番号=2001）
理由説明が異なる　（識別番号=2002）
・推論結果に矛盾する組合せがある
システム影響とファイル影響度が矛盾する　（識別番号=2003, 2004）
・推論結果とデータベース内容が矛盾する
対ディスクの状態が正常で、推論結果に障害の影響がある
（識別番号=2005, 2006, 2007）
対ディスクの状態が障害で、推論結果が影響なしである
（識別番号=2008）
対ディスクの状態が障害で、推論結果に正常を示す理由説明がある。
（識別番号=2009, 2010, 2011）
ファイルの種類と推論結果の理由説明が矛盾する　（識別番号=2012, 2013）
ファイルの補助情報と推論結果の理由説明が矛盾する

（識別番号=2014, 2015, 2016）

(3)rule3に関する矛盾ルール

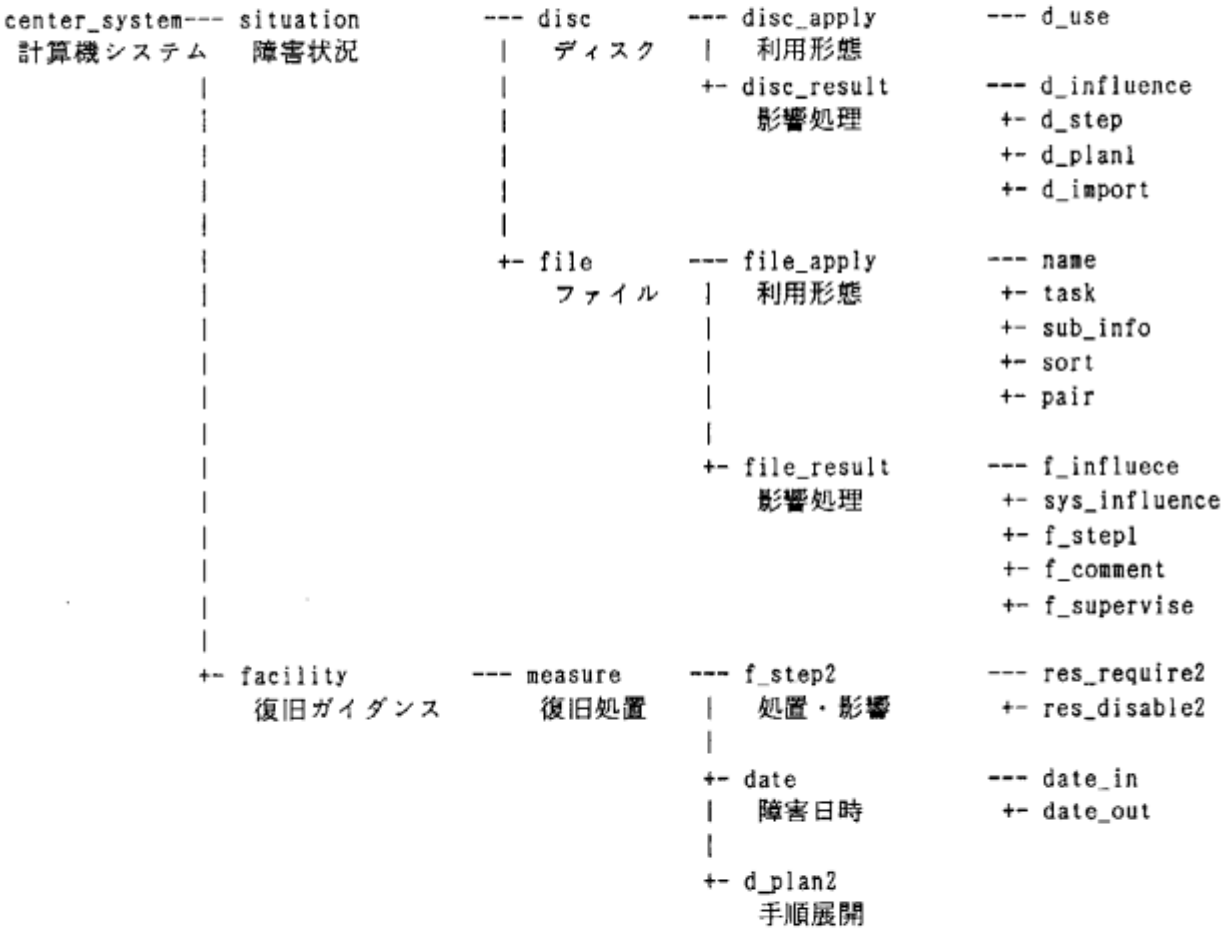
・推論結果の復旧手順が、テストケースの設定と異なる（識別番号=2017）

・推論結果に矛盾する組合せがある。

日時と復旧手順との不一致　（識別番号=2018, 2019）

A.1.4　仮説生成

(1)仮説生成知識（部分）



(2)仮説生成情報

診断ルールの全てのプリミティブ（if部に記述する条件項目、then部に記述する結果項目）に対して、詳細化方向・並列概念・上位概念の各1レベルの項目を仮説生成の対象とする。

B.1.5　テストケース

検証内容

ディスク障害復旧支援用の知識のうち、ファイル単位で影響・処置を判定するrule2について、２重化ファイルを含むディスクの両側の障害に対する知識の不備を検証操作によって修正する。

このため知識ベースに以下の矛盾ルールを設定しておく。

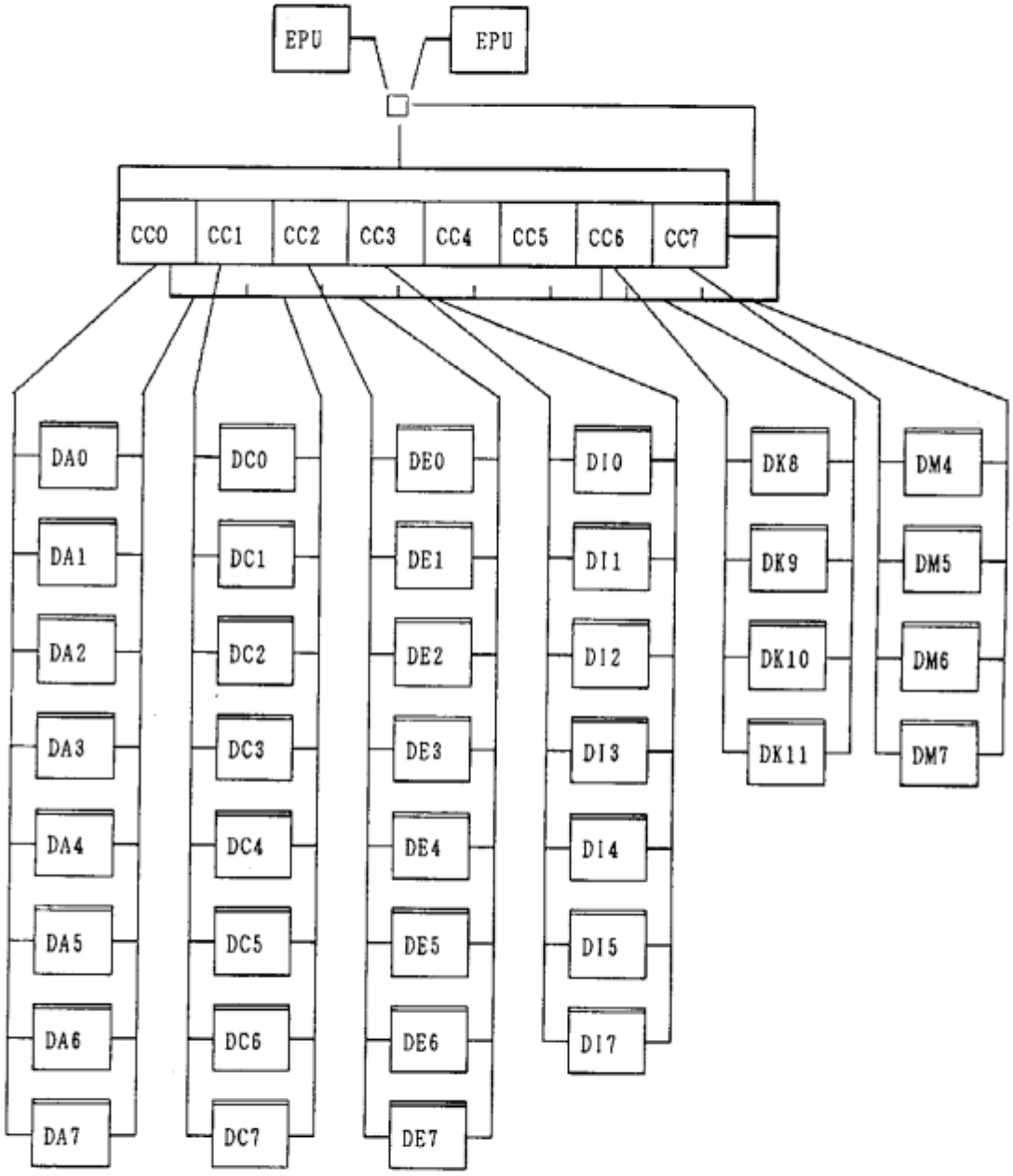
```
rule2(1,10,[FNAME,DISK],[FNAME,DISK,INF,SYSINF,STEP,FREASON,VSOP]):-
    ファイル包含(含む,1,FNAME,['システム構成情報ファイル']),
    対ディスク状態(障害,2,FNAME,DISK,PAIR,LIVE),
×  →ＴＳＳ影響(ファイル障害なし,22,INF),
    システム影響(使用可,1,SYSINF),
    処理影響(セットアップリカバリ不可,224,STEP),
    理由説明(多重化ファイル,22,FREASON),
    ファイル影響(軽微,2,VSOP).
```

検証後の修正目標となる知識は以下に示す通りである。
（rule2の第２引数のXXは、仮説生成時に決められる識別番号である）

```
rule2(1,XX,[FNAME,DISK],[FNAME,DISK,INF,SYSINF,STEP,FREASON,VSOP]):-
    ファイル包含(含む,1,FNAME,['システム構成情報ファイル']),
    対ディスク状態(障害,2,FNAME,DISK,PAIR,LIVE),
◎  →ＴＳＳ影響(ファイル障害影響有り,21,INF),
    システム影響(使用可,1,SYSINF),
    処理影響(セットアップリカバリ不可,224,STEP),
    理由説明(多重化ファイル,22,FREASON),
    ファイル影響(軽微,2,VSOP).
```

"→"印の所が修正された知識が生成される。
本検証例ではこれ以外にも知識が生成される。それらは何れも後述のテストケースを満足しており有っても問題のない知識であるが、これらを知識として追加するか否かは最終的には人間の判断となる。

テストディスク構成



図B-2　テストケースディスク構成

テストデータ

①障害状況

DISK：障害ディスク	FNAME：障害ファイ	ファイル種類
da1,dc1	工程ﾚﾎﾟｰﾄﾌｧｲﾙ ｼｽﾃﾑ構成情報ﾌｧｲﾙ ｺﾝｿｰﾙﾛｸﾞﾌｧｲﾙ	２重 ２重 ２重

②判定出力結果

rule2適用結果

FNAME　：障害ファイル	工程ﾚﾎﾟｰﾄﾌｧｲﾙ	ｼｽﾃﾑ構成情報ﾌｧｲﾙ	ｺﾝｿｰﾙﾛｸﾞﾌｧｲﾙ
INF　：TSS業務影響	TSS業務への影響あり		
SYSINF　：システム影響	システム使用不可	システム使用可	システム使用不可
STEP　：処置・状況	TSS業務続行不可能となります	ｾｯﾄｱｯﾌﾟ/ﾘｶﾊﾞﾘｰ不可能です	TSS業務続行不可能となります
FREASON：理由説明	多重化ファイルのため		
VSOP　：ﾌｧｲﾙ影響度	影響度１	影響度２	影響度１

B.2 知識及びデータ記述

知識及びデータは次に示すファイルに定義されている。

知識構成

知識モジュール情報	rule_io.rio
診断知識フレーム	rf3.frm
矛盾検出ルールフレーム	crf.frm

診断知識

診断ルール	m_rule1.rule
	m_rule2_3.rule
	m_rule3.rule
基本概念（プリミティブ）	m_prim.esp
データベース	m_db.db

検証知識

テストケース	t3.tst
矛盾検出ルール	c1.crule
	c2.crule
	c3.crule
仮説生成知識	assum.inf
仮説生成情報	flag.flg

その他

補助情報表示	cg.esp(クラス名はcg)

```
tepsi916::>fd0>rule_io.rio

rule1(a, 0, "DISK], [DISK1, STEP, INF1, DSUP]),
rule2(a, 0, [FNAME, DISK], [FNAME1, DISK2, INF, SYSINF, STEP, FREASON, VSOP]],
rule3(a, 0, [STEP, WHDAY], [STEP1, WHDAY1, DOP]),
error1(c, 0, [DISK1, DSUP], [RID]),
error2(c, 0, [FNAME1, DISK2, INF, SYSINF, STEP, FREASON, VSOP], [RID]),
error3(c, 0, [STEP1, WHDAY1, DOP], [RID]),
```

```
tepsi916::>fd0>rf3.frm

frame(rule1, ["DISK"], ["STEP", "INF", "DSU"],
          [ディスク用途, 処置影響, 'TSS影響', ディスク種別],
          [[ディスク用途], [処置影響, 'TSS影響', ディスク種別]], m_rule1),
frame(rule2, ["FNAME", "DISK"], ["FNAME", "DISK", "INF", "SYSINF",
          "STEP", "REASON", "VSOP"],
          [ファイル種別, ファイル特性, 業務, ファイル包含, 対ディスク状態,
          'TSS影響', システム影響, 処置影響, 理由説明, ファイル影響グループ],
          [[ファイル種別, ファイル特性, 業務, ファイル包含, 対ディスク状態],
          ['TSS影響', システム影響, 処置影響, 理由説明, ファイル影響グループ]], m_rule2_3,
          assum, flag),
frame(rule3, ["STEP", "WHDAY"], ["DOP"],
          [処置影響, 日時, 手順展開], [[処置影響, 日時], [手順展開]], m_rule3),
```

```
tepsi916::>fd0>crf.frm

frame(error1, ["DISK", "DSU"], ["RID"],
          '$$void', '$$void', c1),
frame(error2, ["FNAME", "DISK", "INF", "SYSINF",
          "STEP", "FREASON", "VSOP"], ["RID"],
          '$$void', '$$void', c2),
frame(error3, ["STEP", "WHDAY", "DOP"], ["RID"],
          '$$void', '$$void', c3),
```

```
tepsl916::>fd0>m_rule1.rule

rule1(1, 1, [DISK], [DISK, STEP, INF, DSU]):-
      ディスク用途(予備パック, 5, DISK, K),
      処置影響(影響なし, 12, STEP),
      'TSS影響' (ディスク傷害影響なし, 12, INF),
      ディスク種類(予備, 3, DSU).
rule1(1, 2, [DISK], [DISK, STEP, INF, DSU]):-
      ディスク用途(本番系, 3, DISK, K),
      処置影響(ホットスタンバイ, 11, STEP),
      'TSS影響' (ディスク傷害影響あり, 11, INF),
      ディスク種類(本番系, 1, DSU).
rule1(1, 3, [DISK], [DISK, STEP, INF, DSU]):-
      ディスク用途(待機系, 4, DISK, K),
      処置影響(待機系ダウン, 13, STEP),
      'TSS影響' (ディスク傷害影響なし, 12, INF),
      ディスク種類(待機系, 2, DSU).
```

```
tepsl916::>[d0>m_rule2_3.rule

rule2(1, 1, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
                STEP, FREASON, VSOP]) :-
      ファイル種類('1重', 11, FNAME, DISK, KIND),
      ファイル特性('セットアップ/ターミネーション不可', 12, FNAME, DISK, STATUS),
      'TSS影響'(ファイル傷害影響あり, 21, INF),
      システム影響(使用可, 1, SYSINF),
      処置影響(パック切換え, 211, STEP),
      理由説明(セットアップターミネーション不可, 12, FREASON),
      ファイル影響(軽微, 2, VSOP).
rule2(1, 2, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
                STEP, FREASON, VSOP]) :-
      ファイル種類('1重', 11, FNAME, DISK, KIND),
      ファイル特性(セットアップ不可, 11, FNAME, DISK, STATUS),
      'TSS影響'(ファイル傷害影響あり, 21, INF),
      システム影響(使用可, 1, SYSINF),
      処置影響(パック切換え, 211, STEP),
      理由説明(セットアップ不可, 11, FREASON),
      ファイル影響(軽微, 2, VSOP).
rule2(1, 3, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
                STEP, FREASON, VSOP]) :-
      ファイル種類(書籍, 211, FNAME, DISK, KIND),
      'TSS影響'(ファイル傷害影響あり, 21, INF),
      システム影響(使用可, 1, SYSINF),
      処置影響(再実行, 215, STEP),
      理由説明(書籍ファイル, 211, FREASON),
      ファイル影響(軽微, 2, VSOP).
rule2(1, 4, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
              STEP, FREASON, VSOP]) :-
      ファイル種類(書籍, 211, FNAME, DISK, KIND),
      ファイル特性('セットアップ/リカバリー不可', 13, FNAME, DISK, STATUS),
      not(業務(共通1, 211, FNAME, DISK, TASK)),
      'TSS影響'(ファイル傷害影響あり, 21, INF),
      システム影響(使用可, 1, SYSINF),
      処置影響(ファイル復元, 212, STEP),
      理由説明(セットアップリカバリー不可, 13, FREASON),
      ファイル影響(軽微, 2, VSOP).
rule2(1, 5, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
              STEP, FREASON, VSOP]) :-
      ファイル種類(プログラム, 222, FNAME, DISK, KIND),
      ファイル特性('セットアップ/リカバリー不可', 13, FNAME, DISK, STATUS),
      業務(共通1, 211, FNAME, DISK, TASK),
      'TSS影響'(ファイル傷害影響あり, 21, INF),
      システム影響(使用可, 1, SYSINF),
      処置影響(ファイル復元, 212, STEP),
      理由説明(セットアップリカバリー不可, 13, FREASON),
      ファイル影響(重大, 1, VSOP).
rule2(1, 6, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
              STEP, FREASON, VSOP]) :-
      ファイル種類(プログラム, 222, FNAME, DISK, KIND),
      ファイル特性(その他, 23, FNAME, DISK, STATUS),
      not(業務(共通1, 211, FNAME, DISK, TASK)),
      'TSS影響'(ファイル傷害影響あり, 21, INF),
      システム影響(使用不可, 2, SYSINF),
      処置影響(パック切換え, 211, STEP),
      理由説明(プログラム, 212, FREASON),
      ファイル影響(軽微, 2, VSOP).
rule2(1, 7, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
              STEP, FREASON, VSOP]) :-
      ファイル種類(プログラム, 222, FNAME, DISK, KIND),
```

```
        ファイル特性（その他，23，FNAME，DISK，STATUS），
        業務（共通1，211，FNAME，DISK，TASK），
        ファイル包含（含む，1，FNAME，[' 工程管理履歴プログラムファイル' ]），
        ' TSS影響' （ファイル障害影響あり，21，INF），
        システム影響（使用不可，2，SYSINF），
        処置影響（業務不可，222，STEP），
        ファイル影響（重大，1，VSOP）．
rule2（1，8，[FNAME，DISK]，[FNAME，DISK，INF，SYSINF，
                STEP，FREASON，VSOP]）：－
        ファイル種類（プログラム，222，FNAME，DISK，KIND），
        ファイル特性（その他，23，FNAME，DISK，STATUS），
        業務（共通1，211，FNAME，DISK，TASK），
        ファイル包含（含まない，2，FNAME，[' 工程管理履歴プログラムファイル' ]），
        ' TSS影響' （ファイル障害影響あり，21，INF），
        システム影響（使用不可，2，SYSINF），
        処置影響（ファイル復元，212，STEP），
        理由説明（プログラム，212，FREASON），
        ファイル影響（重大，1，VSOP）．
rule2（1，9，[FNAME，DISK]，[FNAME，DISK，INF，SYSINF，
                STEP，FREASON，VSOP]）：－
        ファイル種類（' 2重' ，12，FNAME，DISK，KIND），
        ファイル特性（バックリカバリー要，21，FNAME，DISK，STATUS），
        対ディスク状態（正常，1，FNAME，DISK，PAIR，LIVE），
        ' TSS影響' （ファイル障害影響なし，22，INF），
        システム影響（使用可，1，SYSINF），
        処置影響（バックリカバリー，213，STEP），
        理由説明（ペア状態によりセットアップリカバリー不可，31，FREASON），
        ファイル影響（軽微，2，VSOP）．
rule2（1，10，[FNAME，DISK]，[FNAME，DISK，INF，SYSINF，
                STEP，FREASON，VSOP]）：－
        ファイル包含（含む，1，FNAME，[' システム構成情報ファイル' ]），
        対ディスク状態（障害，2，FNAME，DISK，PAIR，LIVE），
        ' TSS影響' （ファイル障害影響なし，22，INF），
        システム影響（使用可，1，SYSINF），
        処置影響（セットアップリカバリー不可，224，STEP），
        理由説明（多重化ファイル，22，FREASON），
        ファイル影響（軽微，2，VSOP）．
rule2（1，11，[FNAME，DISK]，[FNAME，DISK，INF，SYSINF，
                STEP，FREASON，VSOP]）：－
        ファイル種類（課金，221，FNAME，DISK，KIND），
        対ディスク状態（正常，1，FNAME，DISK，PAIR，LIVE），
        ' TSS影響' （ファイル障害影響あり，21，INF），
        システム影響（使用不可，2，SYSINF），
        処置影響（パック切換え，211，STEP），
        理由説明（ペア状態によりTSSに影響あり，32，FREASON），
        ファイル影響（軽微，2，VSOP）．
rule2（1，12，[FNAME，DISK]，[FNAME，DISK：INF，SYSINF，
                STEP，FREASON，VSOP]）：－
        ファイル種類（' 2重' ，12，FNAME，DISK，KIND），
        ファイル特性（' DVF切り換え要' ，22，FNAME，DISK，STATUS），
        対ディスク状態（障害，2，FNAME，DISK，PAIR，LIVE），
        ' TSS影響' （ファイル障害影響あり，21，INF），
        システム影響（使用不可，2，SYSINF），
        処置影響（パック切換え，211，STEP），
        理由説明（多重化ファイル，22，FREASON），
        ファイル影響（重大，1，VSOP）．
rule2（1，13，[FNAME，DISK]，[FNAME，DISK，INF，SYSINF，
                STEP，FREASON，VSOP]）：－
        ファイル種類（' 2重' ，12，FNAME，DISK，KIND），
        ファイル特性（その他，23，FNAME，DISK，STATUS），
        対ディスク状態（正常，1，FNAME，DISK，PAIR，LIVE），
        ' TSS影響' （ファイル障害影響なし，22，INF），
        システム影響（使用可，1，SYSINF），
        処置影響（処置不明，217，STEP），
        理由説明（多重化ファイル，22，FREASON），
        ファイル影響（軽微，2，VSOP）．
```

```
rule2(1, 11, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
        STEP, FREASON, VSOP]) :-
    ファイル種類('2重', 12, FNAME, DISK, KIND),
    ファイル特性(その他, 23, FNAME, DISK, STATUS),
    業務(文書管理, 14, FNAME, DISK, TASK),
    対ディスク状態(傷害, 2, FNAME, DISK, PAIR, LIVE),
    'TSS影響'(ファイル傷害影響あり, 21, INF),
    システム影響(使用不可, 2, SYSINF),
    処置影響(文書管理不可, 223, STEP),
    理由説明(ペア状態によりシステム傷害, 33, FREASON),
    ファイル影響(重大, 1, VSOP).
rule2(1, 15, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
        STEP, FREASON, VSOP]) :-
    ファイル種類('2重', 12, FNAME, DISK, KIND),
    ファイル特性(その他, 23, FNAME, DISK, STATUS),
    対ディスク状態(傷害, 2, FNAME, DISK, PAIR, LIVE),
    業務(原価管理, 13, FNAME, DISK, TASK),
    'TSS影響'(ファイル傷害影響あり, 21, INF),
    システム影響(使用可, 1, SYSINF),
    処置影響(発注不可, 225, STEP),
    理由説明(多重化ファイル, 22, FREASON),
    ファイル影響(重大, 1, VSOP).
rule2(1, 17, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
        STEP, FREASON, VSOP]) :-
    ファイル種類('2重', 12, FNAME, DISK, KIND),
    ファイル特性(その他, 23, FNAME, DISK, STATUS),
    対ディスク状態(傷害, 2, FNAME, DISK, PAIR, LIVE),
    業務(連絡, 15, FNAME, DISK, TASK),
    'TSS影響'(ファイル傷害影響あり, 21, INF),
    システム影響(使用不可, 2, SYSINF),
    処置影響(フォールバック, 216, STEP),
    理由説明(ペア状態によりシステム傷害, 33, FREASON),
    ファイル影響(重大, 1, VSOP).
rule2(1, 18, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
        STEP, FREASON, VSOP]) :-
    ファイル種類('2重', 12, FNAME, DISK, KIND),
    ファイル特性(その他, 23, FNAME, DISK, STATUS),
    業務(共通1, 211, FNAME, DISK, TASK),
    対ディスク状態(傷害, 2, FNAME, DISK, PAIR, LIVE),
    ファイル包含(含まない, 2, FNAME, ['工程レポートファイル']),
    ファイル包含(含まない, 2, FNAME, ['客先仕様履歴ファイル', '設計仕様履歴ファイル']),
    ファイル包含(含まない, 2, FNAME, ['系立ち上げ制御ファイル', '系切離し制御ファイル']),
    ファイル包含(含まない, 2, FNAME, ['仕様マスタファイル']),
    'TSS影響'(ファイル傷害影響あり, 21, INF),
    システム影響(使用不可, 2, SYSINF),
    処置影響(業務不可, 222, STEP),
    理由説明(多重化ファイル, 22, FREASON),
    ファイル影響(重大, 1, VSOP).
rule2(1, 19, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
        STEP, FREASON, VSOP]) :-
    ファイル包含(含む, 1, FNAME, ['工程レポートファイル']),
    対ディスク状態(傷害, 2, FNAME, DISK, PAIR, LIVE),
    'TSS影響'(ファイル傷害影響あり, 21, INF),
    システム影響(使用不可, 2, SYSINF),
    処置影響(業務不可, 222, STEP),
    理由説明(多重化ファイル, 22, FREASON),
    ファイル影響(重大, 1, VSOP).
rule2(1, 20, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
        STEP, FREASON, VSOP]) :-
    対ディスク状態(傷害, 2, FNAME, DISK, PAIR, LIVE),
    ファイル包含(含む, 1, FNAME, ['客先仕様履歴ファイル', '設計仕様履歴ファイル']),
    'TSS影響'(ファイル傷害影響あり, 21, INF),
    システム影響(使用不可, 2, SYSINF),
    処置影響(スケジューリング不可, 221, STEP),
    理由説明(多重化ファイル, 22, FREASON),
    ファイル影響(重大, 1, VSOP).
```

```
rule2(1, 21, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
        STEP, FREASON, VSOP]) :-
    対ディスク状態(障害, 2, FNAME, DISK, PAIR, LIVE),
    ファイル包含(含む, 1, FNAME, ['系立ち上げ制御ファイル', '系切離し制御ファイル']),
    'TSS影響'(ファイル障害影響あり, 21, INF),
    システム影響(使用不可, 2, SYSINF),
    処置影響(スケジューリング不可, 221, STEP),
    理由説明(多重化ファイル, 22, FREASON),
    ファイル影響(重大, 1, VSOP).
rule2(1, 22, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
        STEP, FREASON, VSOP]) :-
    対ディスク状態(障害, 2, FNAME, DISK, PAIR, LIVE),
    ファイル包含(含む, 1, FNAME, ['仕様マスタファイル']),
    'TSS影響'(ファイル障害影響あり, 21, INF),
    システム影響(使用不可, 2, SYSINF),
    処置影響(フォールバック, 216, STEP),
    理由説明(多重化ファイル, 22, FREASON),
    ファイル影響(重大, 1, VSOP).
rule2(1, 23, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
        STEP, FREASON, VSOP]) :-
    ファイル種類('3重', 13, FNAME, DISK, KIND),
    対ディスク状態(障害, 2, FNAME, DISK, PAIR, LIVE),
    'TSS影響'(ファイル障害影響あり, 21, INF),
    システム影響(使用不可, 2, SYSINF),
    処置影響(業務不可, 222, STEP),
    理由説明(多重化ファイル, 22, FREASON),
    ファイル影響(重大, 1, VSOP).
rule2(1, 24, [FNAME, DISK], [FNAME, DISK, INF, SYSINF,
        STEP, FREASON, VSOP]) :-
    ファイル種類('3重', 13, FNAME, DISK, KIND),
    対ディスク状態(正常, 1, FNAME, DISK, PAIR, LIVE),
    'TSS影響'(ファイル障害影響なし, 22, INF),
    システム影響(使用可, 1, SYSINF),
    処置影響(処置不明, 217, STEP),
    理由説明(多重化ファイル, 22, FREASON),
    ファイル影響(軽微, 2, VSOP).
```

```
tepsi>lo:::\rd0>m_rule3.rule

rule3(1, 1, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(ホットスタンバイ, 11, STEP),
      日時(平日時間外, 12, WHDAY),
      手順展開(連絡し復元, 12, DOP).
rule3(1, 2, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(ホットスタンバイ, 11, STEP),
      日時(土曜時間外, 22, WHDAY),
      手順展開(連絡し復元, 12, DOP).
rule3(1, 3, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(ホットスタンバイ, 11, STEP),
      日時(平日時間内, 11, WHDAY),
      手順展開(連絡, 11, DOP).
rule3(1, 4, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(ホットスタンバイ, 11, STEP),
      日時(土曜時間内, 21, WHDAY),
      手順展開(連絡, 11, DOP).
rule3(1, 5, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(影響なし, 12, STEP),
      手順展開(バックチェックし復元, 3, DOP).
rule3(1, 6, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(フォールバック, 216, STEP),
      日時(土曜時間外, 22, WHDAY),
      手順展開(バックチェック, 2, DOP).
rule3(1, 7, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(フォールバック, 216, STEP),
      日時(平日時間外, 12, WHDAY),
      手順展開(連絡し復元, 12, DOP).
rule3(1, 8, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(フォールバック, 216, STEP),
      日時(土曜時間内, 21, WHDAY),
      手順展開(連絡, 11, DOP).
rule3(1, 9, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(フォールバック, 216, STEP),
      日時(平日時間内, 11, WHDAY),
      手順展開(連絡, 11, DOP).
rule3(1, 10, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(バックリカバリー, 213, STEP),
      手順展開(バックチェック, 2, DOP).
rule3(1, 11, [STEP, WHDAY], [STEP, WHDAY, DOP]):-
      処置影響(バック切換え, 211, STEP),
      日時(平日時間外, 12, WHDAY),
      手順展開(バックチェックし復元, 3, DOP).
```

```
tepsi016:::fd0>m_prim.esp

class    m_prim   has

attribute
      atms
      ;

:ディスク用途(_,DB,生産管理,1,DISK,K):-
      :db(DB,disk,DISK,'生産管理サブシステム',_),
      K='生産管理サブシステム';
:ディスク用途(_,DB,原価管理,2,DISK,K):-
      :db(DB,disk,DISK,原価管理サブシステム,_),
      K=原価管理サブシステム;
:ディスク用途(_,DB,本番系,3,DISK,K):-
      :db(DB,disk,DISK,本番系SYS,_),
      K=本番系システム;
:ディスク用途(_,DB,待機系,4,DISK,K):-
      :db(DB,disk,DISK,待機系SYS,_),
      K=待機系システム;
:ディスク用途(_,DB,予備パック,5,DISK,K):-
      :db(DB,disk,DISK,予備パック,_),
      K=予備パック;

:日時(_,_,平日時間内,11,[DAY,TIME]):-
      date(1,DAY),
      time(1,TIME);
:日時(_,_,平日時間外,12,[DAY,TIME]):-
      date(1,DAY),
      not(time(1,TIME));
:日時(_,_,土曜時間内,21,[DAY,TIME]):-
      date(2,DAY),
      time(2,TIME);
:日時(_,_,土曜時間外,22,[DAY,TIME]):-
      date(2,DAY),
      not(time(2,TIME));
:日時(_,_,日曜,3,[DAY,_],_):-
      date(3,DAY);

:処置影響(_,_,ホットスタンバイ,11,STEP):-
      STEP=本番系に付きホットスタンバイ実施;
:処置影響(_,_,影響なし,12,STEP):-
      STEP=予備パックに付き影響なし;
:処置影響(_,_,待機系ダウン,13,STEP):-
      STEP=待機系システムダウン;
:処置影響(_,_,業務使用中,14,STEP):-
      STEP=このディスクは使用中;
:処置影響(_,_,不明,15,STEP):-
      STEP=[];
:処置影響(_,_,パック切換え,211,STEP):-
      STEP=予備パックへの切り換え必要;
:処置影響(_,_,ファイル復元,212,STEP):-
      STEP=予備パックにファイルの復元必要;
:処置影響(_,_,バックリカバリー,213,STEP):-
      STEP=バックリカバリーが必要;
:処置影響(_,_,'MTバックアップ',214,STEP):-
      STEP=トランザクション途中のMTに復旧;
:処置影響(_,_,再実行,215,STEP):-
      STEP=処理リランが必要;
:処置影響(_,_,フォールバック,216,STEP):-
      STEP=フォールバックオールリカバリーが必要;
:処置影響(_,_,処置不明,217,STEP):-
```

```
        STEP＝特別な処理は不要です；
:処理影響（_，_，スケジューリング不可，２２１，STEP）：－
        STEP＝ジョブスケジューリング不可能；
:処理影響（_，_，業務不可，２２２，STEP）：－
        STEP＝オンライン続行不可能となります；
:処理影響（_，_，文書管理不可，２２３，STEP）：－
        STEP＝ドキュメント管理システムが続行不可能になります；
:処理影響（_，_，セットアップリカバリー不可，２２４，STEP）：－
        STEP＝'セットアップ／リカバリー不可能です'；
:処理影響（_，_，発注不可，２２５，STEP）：－
        STEP＝'発注，'完成処理が不可能です'；

:'ＴＳＳ影響'（_，_，ディスク傷害影響あり，１１，INF）：－
        INF＝ディスクによるオンラインへの影響あり；
:'ＴＳＳ影響'（_，_，ディスク傷害影響なし，１２，INF）：－
        INF＝ディスクによるオンラインへの影響なし；
:'ＴＳＳ影響'（_，_，ファイル傷害影響あり，２１，INF）：－
        INF＝ファイルによるオンラインへの影響あり；
:'ＴＳＳ影響'（_，_，ファイル傷害影響なし，２２，INF）：－
        INF＝ファイルによるオンラインへの影響なし；

:手順展開（_，_，連絡，１１，DOP）：－
        DOP＝'オペレータを配置してください．次の関係各課へ連絡してください．'；
:手順展開（_，_，連絡し復元，１２，DOP）：－
        DOP＝'オペレータを配置してください．次の関係各課へ連絡してください．メディアを復元してください．'；
:手順展開（_，_，バックチェック，２，DOP）：－
        DOP＝'ＪＣＬを用意してください．バックアップデータを用意してください．予備パックのチェックをしてください．'；
:手順展開（_，_，バックチェックし復元，３，DOP）：－
        DOP＝'ＪＣＬを用意してください．バックアップデータを用意してください．予備パックのチェックをしてください．次の復旧操作を実施してください'；

:ディスク種類（_，_，本番系，１，DSU）：－
        DSU＝本番系システムパック；
:ディスク種類（_，_，待機系，２，DSU）：－
        DSU＝待機系システムパック；
:ディスク種類（_，_，予備，３，DSU）：－
        DSU＝予備パック；
:ディスク種類（_，_，その他，４，DSU）：－
        DSU＝その他；

:ファイル種類（_，DB，'１重'，１１，FNAME，DISK，KIND）：－
        :db（DB，file，FNAME，DISK，_，'１重'，_），
        KIND＝'１重'；
:ファイル種類（_，DB，'２重'，１２，FNAME，DISK，KIND）：－
        :db（DB，file，FNAME，DISK，_，'２重'，_），
        KIND＝'２重'；
:ファイル種類（_，DB，'３重'，１３，FNAME，DISK，KIND）：－
        :db（DB，file，FNAME，DISK，_，'３重'，_），
        KIND＝'３重'；
:ファイル種類（_，DB，蓄積，２１１，FNAME，DISK，KIND）：－
        :db（DB，file，FNAME，DISK，_，蓄積データ，_），
        KIND＝蓄積データ；
:ファイル種類（_，DB，課金，２２１，FNAME，DISK，KIND）：－
        :db（DB，file，FNAME，DISK，_，課金データ，_），
        KIND＝課金データ；
:ファイル種類（_，DB，ジャーナル，２１２，FNAME，DISK，KIND）：－
        :db（DB，file，FNAME，DISK，_，ジャーナル，_），
        KIND＝ジャーナル；
:ファイル種類（_，DB，プログラム，２２２，FNAME，DISK，KIND）：－
        :db（DB，file，FNAME，DISK，_，プログラム，_），
        KIND＝プログラム；
:ファイル種類（_，DB，不明，３，FNAME，DISK，KIND）：－
        :db（DB，file，FNAME，DISK，_，_，_），
        KIND＝不明；
:ファイル種類（_，DB，等しい，９９，FNAME，DISK，KIND）：－
        :db（DB，file，FNAME，DISK，_，KIND，_）
        ；
```

```
:業務(_, DB, 品質管理, 11, FNAME, DISK, TASK) :-
      :db(DB, file, FNAME, DISK, '品質管理問い合わせ_更新', _, _),
      TASK='品質管理問い合わせ_更新';
:業務(_, DB, 工程管理, 12, FNAME, DISK, TASK) :-
      :db(DB, file, FNAME, DISK, '工程管理問い合わせ_更新', _, _),
      TASK='工程管理問い合わせ_更新';
:業務(_, DB, 原価管理, 13, FNAME, DISK, TASK) :-
      :db(DB, file, FNAME, DISK, '原価管理問い合わせ_更新', _, _),
      TASK='原価管理問い合わせ_更新';
:業務(_, DB, 文書管理, 14, FNAME, DISK, TASK) :-
      :db(DB, file, FNAME, DISK, ドキュメント管理, _, _),
      TASK=ドキュメント管理;
:業務(_, DB, 連絡, 15, FNAME, DISK, TASK) :-
      :db(DB, file, FNAME, DISK, 工場間連絡, _, _),
      TASK=工場間連絡;
:業務(_, DB, 共通1, 211, FNAME, DISK, TASK) :-
      :db(DB, file, FNAME, DISK, 共通1, _, _),
      TASK=共通1;
:業務(_, DB, 共通2, 221, FNAME, DISK, TASK) :-
      :db(DB, file, FNAME, DISK, 共通2, _, _),
      TASK=共通2;
:業務(_, DB, 課金, 212, FNAME, DISK, TASK) :-
      :db(DB, file, FNAME, DISK, 課金, _, _),
      TASK=課金;

:ファイル特性(_, DB, セットアップ不可, 11, FNAME, DISK, STATUS) :-
      :db(DB, file, FNAME, DISK, _, _, セットアップ不可),
      STATUS=セットアップ不可;
:ファイル特性(_, DB, 'セットアップ_ターミネーション不可', 12,
            FNAME, DISK, STATUS) :-
      :db(DB, file, FNAME, DISK, _, _, 'セットアップ/ターミネーション不可'),
      STATUS='セットアップ/ターミネーション不可';
:ファイル特性(_, DB, 'セットアップ/リカバリー不可', 13, FNAME, DISK, STATUS) :-
      :db(DB, file, FNAME, DISK, _, _, 'セットアップ/リカバリー/ターミネーション不可'),
      STATUS='セットアップ_リカバリー/ターミネーション不可';
:ファイル特性(_, DB, バックリカバリー要, 21, FNAME, DISK, STATUS) :-
      :db(DB, file, FNAME, DISK, _, _, バックリカバリー要),
      STATUS=バックリカバリー要;
:ファイル特性(_, DB, 'DVF切り換え要', 22, FNAME, DISK, STATUS) :-
      :db(DB, file, FNAME, DISK, _, _, 'DVF切り換え要'),
      STATUS='DVF切り換え要';
:ファイル特性(_, DB, その他, 23, FNAME, DISK, STATUS) :-
      :db(DB, file, FNAME, DISK, _, _, その他),
      STATUS=その他;

:ファイル包含(_, _, 含む, 1, FNAME1, FNAME2) :-
      member(FNAME1, FNAME2);
:ファイル包含(_, _, 含まない, 2, FNAME1, FNAME2) :-
      not(member(FNAME1, FNAME2));

:対ディスク状態(_, DB, 正常, 1, FNAME, DISK, PAIR, LIVE) :-
      :db(DB, pair, FNAME, PAIR),
      one_of(One, PAIR),
      One\==DISK,
      :db(DB, disk, One, _, live),
      LIVE=live;
:対ディスク状態(_, DB, 障害, 2, FNAME, DISK, PAIR, LIVE) :-
      :db(DB, pair, FNAME, PAIR),
      ( one_of(One, PAIR),
        One\==DISK,
        :db(DB, disk, One, _, live),
        cut_and_fail;
        LIVE=die );

:システム影響(_, _, 使用可, 1, SYSINF) :-
```

```
        SYSINF=システム使用可;
:システム影響(_, _, 使用不可, 2, SYSINF):-
        SYSINF=システム使用不可;

:理由説明(_, _, セットアップ不可, 11, REASON):-
        REASON=セットアップ処理が不可となるため;
:理由説明(_, _, セットアップターミネーション不可, 12, REASON):-
        REASON='セットアップ/ターミネーション処理が不可となるため';
:理由説明(_, _, セットアップリカバリー不可, 13, REASON):-
        REASON='セットアップ/リカバリー/ターミネーション処理が不可となるため';
:理由説明(_, _, 特殊ファイル, 21, REASON):-
        REASON='特殊ファイルのため';
:理由説明(_, _, 多重化ファイル, 22, REASON):-
        REASON='多重化ファイルのため';
:理由説明(_, _, 蓄積ファイル, 211, REASON):-
        REASON='蓄積ファイルのため';
:理由説明(_, _, プログラム, 212, REASON):-
        REASON='プログラムファイルのため';
:理由説明(_, _, ペア状態によりセットアップリカバリー不可, 31, REASON):-
        REASON='ペアディスクも障害になるとセットアップ/リカバリーが不可能になるため';
:理由説明(_, _, ペア状態によりTSSに影響あり, 32, REASON):-
        REASON='ペアディスクも障害になるとオンラインへの影響があるため';
:理由説明(_, _, ペア状態によりシステム障害, 33, REASON):-
        REASON='ペアディスクも障害になるとシステムが障害となるため';

:ファイル影響(_, _, 重大, 1, VSOP):-
        VSOP=ファイル影響度1;
:ファイル影響(_, _, 軽微, 2, VSOP):-
        VSOP=ファイル影響度2;

:テストケース(_, DB, 日時, 1, X, Y):-
        :db(DB, test, date, X, Y)
        ;
:テストケース(_, DB, テストケースの入出力, 3, X, Y):-
        :db(DB, test, all, X, Y)
        ;

:操作(_, _, 等しい, 11, X, _):-
        unbound(X),
        cut_and_fail
        ;
:操作(_, _, 等しい, 11, _, X):-
        unbound(X),
        cut_and_fail
        ;
:操作(_, _, 等しい, 11, X, Y):-
        string(X, _, _), !,
        equal_string(X, Y)
        ;
:操作(_, _, 等しい, 11, X, X);
:操作(_, _, 異なる, 12, X, _):-
        unbound(X),
        cut_and_fail
        ;
:操作(_, _, 異なる, 12, _, X):-
        unbound(X),
        cut_and_fail
        ;
:操作(_, _, 異なる, 12, X, Y):-
        string(X, _, _), !,
        not(equal_string(X, Y))
        ;
:操作(_, _, 異なる, 12, X, Y):-
        X \= Y
        ;
:操作(_, _, インバー, 2, X, Y):-
```

```
        member(X, Y)
        ;

local

        date(1, DAY) :-
             member(DAY, [mon, tue, wed, thu, fri]);
        date(2, sat) ;
        date(3, sun)
             ;

        time(1, TIME) :-
             TIME>=800,
             TIME=<1900;
        time(2, TIME) :-
             TIME>=800,
             TIME=<1200
             ;

        live(_, _, []);
        live(DB, DISK, [One|Rest]) :-
             One=DISK,
             live(DB, DISK, Rest);
        live(DB, DISK, [One|Rest]) :-
             :db(DB, disk, One, _, live),
             live(DB, DISK, Rest)
             ;

        member(X, [X|_]) :-!;
        member(X, [_|L]) :-
             member(X, L);

        one_of(X, [X|_]);
        one_of(X, [_|L]) :-
             one_of(X, L);

end.
```

```
tepsi916::~fd0~m_db.db

db(disk,da0,生産管理サブシステム,live).
db(disk,da1,生産管理サブシステム,live).
db(disk,da2,生産管理サブシステム,live).
db(disk,da3,生産管理サブシステム,live).
db(disk,da4,生産管理サブシステム,live).
db(disk,da5,生産管理サブシステム,live).
db(disk,da6,生産管理サブシステム,live).
db(disk,da7,生産管理サブシステム,live).
db(disk,dc0,生産管理サブシステムlive).
db(disk,dc1,生産管理サブシステム,live).
db(disk,dc2,生産管理サブシステム,live).
db(disk,dc3,生産管理サブシステム,live).
db(disk,dc4,生産管理サブシステム,live).
db(disk,dc5,生産管理サブシステム,live).
db(disk,dc6,生産管理サブシステム,live).
db(disk,dc7,生産管理サブシステム,live).
db(disk,dk8,生産管理サブシステム,live).
db(disk,dk9,生産管理サブシステム,live).
db(disk,dk10,予備,live).
db(disk,dk11,生産管理サブシステム,live).
db(disk,dm4,生産管理サブシステム,live).
db(disk,dm5,生産管理サブシステム,live).
db(disk,dm6,予備,live).
db(disk,dm7,予備,live).
db(disk,de3,生産管理サブシステム,live).
db(disk,de7,生産管理サブシステム,live).
db(disk,di3,生産管理サブシステム,live).
db(disk,di7,生産管理サブシステム,live).
db(disk,de0,本番系SYS,live).
db(disk,de1,本番系SYS,live).
db(disk,de2,本番系SYS,live).
db(disk,de4,本番系SYS,live).
db(disk,de5,本番系SYS,live).
db(disk,de6,本番系SYS,live).
db(disk,di0,待機系SYS,live).
db(disk,di1,待機系SYS,live).
db(disk,di2,待機系SYS,live).
db(disk,di4,待機系SYS,live).
db(disk,di5,待機系SYS,live).
db(file,当日待避関連ファイル,da0,'工程管理問い合わせ/更新';'1重','セットアップ/ターミネーション不可').
db(file,部門情報ファイル,da0,'工程管理問い合わせ/更新','1重',セットアップ不可).
db(file,工程管理ディスクジャーナル,da0,'工程管理問い合わせ/更新',蓄積データ,その他).
db(file,マスタースケジュールファイル,da0,共通1,'1重',セットアップ不可).
db(file,工程レポートファイル,da1,共通1,'2重',その他).
db(file,システム構成情報ファイル,da1,共通2,'2重',バックリカバリー要).
db(file,コンソールログファイル,da1,共通1,'2重',その他).
db(file,システム設計実績ファイル,da2,'工程管理問い合わせ/更新','2重',その他).
db(file,システム設計当日ファイル,da2,'工程管理問い合わせ/更新','2重',その他).
db(file,予算実績ファイル,da2,共通1,'2重','DVF切り換え要').
db(file,影響診断JCLファイル,da2,共通1,'プログラム','セットアップ/リカバリー/ターミネーション/パッチ不可').
db(file,系切り換え制御ファイル,da2,共通1,'2重',その他).
db(file,仕様マスタファイル,da2,共通1,'2重',その他).
db(file,実績能合マスタファイル,da3,共通1,'3重','DVF切り換え要').
db(file,プログラム設計実績ファイル,da4,'工程管理問い合わせ/更新','2重',その他).
db(file,プログラム設計当日ファイル,da4,'工程管理問い合わせ/更新','2重',その他).
db(file,コーディング実績ファイル,da5,'工程管理問い合わせ/更新','2重',その他).
db(file,プロジェクトスケジューリングファイル,da5,'工程管理問い合わせ/更新','2重',その他).
db(file,コーディング当日ファイル,da5,'工程管理問い合わせ/更新','2重',その他).
db(file,製造実績ファイル,da6,'工程管理問い合わせ/更新','2重',その他).
db(file,工場間連絡ファイル,da7,'工場間連絡','2重',その他).
```

```
db(file, ドキュメント管理ファイル, dc7, ドキュメント管理, '2重', その他).
db(file, 原価管理用ファイル, dc7, '原価管理問い合わせ/更新', '2重', その他).
db(file, 下位互換プログラムファイル, dc0, 共通2, プログラム, 'セットアップ/リカバリー/ターミネーション/バッチ不可').
db(file, バッチプログラムファイル, dc0, '工場間連絡', プログラム, 'セットアップ/リカバリー/ターミネーション/バッチ不可').
db(file, 工程参照プログラムファイル, dc0, 共通1, プログラム, 'セットアップ/リカバリー/ターミネーション/バッチ不可').
db(file, 復旧JCLファイル, dc0, 共通1, プログラム, 'セットアップ/リカバリー/ターミネーション/バッチ不可').
db(file, 工程管理ディスクジャーナルファイル, dc0, '工程管理問い合わせ/更新', 蓄積データ, その他).
db(file, 'EDPプログラムファイル', dc0, '原価管理問い合わせ/更新', プログラム, 'セットアップ/リカバリー/ターミネーション/バッチ不可').
db(file, ユーザープログラムファイル, dc0, 共通2, プログラム, 'セットアップ/リカバリー/ターミネーション/バッチ不可').
db(file, システム構成情報ファイル, dc1, 共通2, '2重', バックリカバリー要).
db(file, コンソールログファイル, dc1, 共通1, '2重', その他).
db(file, 工程レポートファイル, dc1, 共通1, '2重', その他).
db(file, システム設計実績ファイル, dc2, '工程管理問い合わせ/更新', '2重', その他).
db(file, システム設計当日ファイル, dc2, '工程管理問い合わせ/更新', '2重', その他).
db(file, 予算実績ファイル, dc2, 共通1, '2重', 'DVF切り換え要').
db(file, 系切り換え制御ファイル, dc2, 共通1, '2重', その他).
db(file, 仕様マスタファイル, dc2, 共通1, '2重', その他).
db(file, 実績総合マスタファイル, dc3, 共通1, '3重', 'DVF切り換え要').
db(file, プログラム設計実績ファイル, dc4, '工程管理問い合わせ/更新', '2重', その他).
db(file, プログラム設計当日ファイル, dc4, '工程管理問い合わせ/更新', '2重', その他).
db(file, プロジェクトスケジューリングファイル, dc5, '工程管理問い合わせ/更新', '2重', その他).
db(file, コーディング当日ファイル, dc5, '工程管理問い合わせ/更新', '2重', その他).
db(file, コーディング実績ファイル, dc5, '工程管理問い合わせ/更新', '2重', その他).
db(file, 製造実績ファイル, dc6, '工程管理問い合わせ/更新', '2重', その他).
db(file, 工場間連絡ファイル, dc7, 工場間連絡, '2重', その他).
db(file, ドキュメント管理ファイル, dc7, ドキュメント管理, '2重', その他).
db(file, 原価管理用ファイル, dc7, '原価管理問い合わせ/更新', '2重', その他).
db(file, 検査実績ファイル, dk8, '工程管理問い合わせ/更新', '2重', その他).
db(file, 検査当日ファイル, dk9, '工程管理問い合わせ/更新', '2重', その他).
db(file, 実績総合マスタファイル, dk11, 共通1, '3重', 'DVF切り換え要').
db(file, 検査実績ファイル, dm4, '工程管理問い合わせ/更新', '2重', その他).
db(file, 検査当日ファイル, dm5, '工程管理問い合わせ/更新', '2重', その他).
db(file, 系立ち上げ制御ファイル, de3, 共通1, '2重', その他).
db(file, 客先仕様履歴ファイル, de3, 共通1, '2重', その他).
db(file, 初期化JCLファイル, de7, 共通1, プログラム, 'セットアップ/リカバリー/ターミネーション/バッチ不可').
db(file, 復旧プログラムファイル, de7, 共通1, プログラム, 'セットアップ/リカバリー/ターミネーション/バッチ不可').
db(file, 工程管理復旧プログラムファイル, de7, 共通1, '2重', その他).
db(file, 系立ち上げ制御ファイル, de7, 共通1, '2重', その他).
db(file, 客先仕様履歴ファイル, de7, 共通1, '2重', その他).
db(file, 端末制御プログラムファイル, de7, 共通1, プログラム, その他).
db(file, アクセス管理プログラムファイル, de7, 共通1, プログラム, その他).
db(file, 系切離し制御ファイル, di3, 共通1, '2重', その他).
db(file, 設計仕様履歴ファイル, di3, 共通1, '2重', その他).
db(file, 復旧完了JCLファイル, di7, 共通1, プログラム, 'セットアップ/リカバリー/ターミネーション/バッチ不可').
db(file, 復旧プログラムファイル, di7, 共通1, プログラム, 'セットアップ/リカバリー/ターミネーション/バッチ不可').
db(file, 工程管理復旧プログラムファイル, di7, 共通1, プログラム, その他).
db(file, 系切離し制御ファイル, di7, 共通1, '2重', その他).
db(file, 設計仕様履歴ファイル, di7, 共通1, '2重', その他).
db(file, 端末制御プログラムファイル, di7, 共通1, プログラム, その他).
db(file, アクセス管理プログラムファイル, di7, 共通1, プログラム, その他).
db(pair, 翌日時越関連ファイル, []).
db(pair, 部門情報ファイル, []).
db(pair, 工程管理ディスクジャーナル, []).
db(pair, マスタースケジュールファイル, []).
db(pair, 工程レポートファイル, [da1, dc1]).
db(pair, システム構成情報ファイル, [da1, dc1]).
db(pair, コンソールログファイル, [da1, dc1]).
db(pair, システム設計実績ファイル, [da2, dc2]).
db(pair, システム設計当日ファイル, [da2, dc2]).
db(pair, 予算実績ファイル, [da2, dc2]).
db(pair, 影響診断JCLファイル, []).
db(pair, 系切り換え制御ファイル, [da2, dc2]).
db(pair, 仕様マスタファイル, [da2, dc2]).
db(pair, 実績総合マスタファイル, [da3, dc3, dk11]).
db(pair, プログラム設計実績ファイル, [da1, dc1]).
db(pair, プログラム実績ファイル, [da1, dc1]).
```

```
db(pair,コーディング実績ファイル,[da5,dc5]).
db(pair,プロジェクトスケジューリングファイル,[da5,dc5]).
db(pair,コーディング当日ファイル,[da5,dc5]).
db(pair,製造実績ファイル,[da6,dc6]).
db(pair,工場間連絡ファイル,[da7,dc7]).
db(pair,ドキュメント管理ファイル,[da7,dc7]).
db(pair,原価管理用ファイル,[da7,dc7]).
db(pair,'TSSプログラムファイル',[]).
db(pair,バッチプログラムファイル,[]).
db(pair,工程参照プログラムファイル,[]).
db(pair,復旧JCLファイル,[]).
db(pair,工程管理ディスクジャーナルファイル,[]).
db(pair,'EDPプログラムファイル',[]).
db(pair,ユーザープログラムファイル,[]).
db(pair,検査実績ファイル,[dk8,dm4]).
db(pair,検査当日ファイル,[dk9,dm5]).
db(pair,系立ち上げ制御ファイル,[de3,de7]).
db(pair,客先仕様履歴ファイル,[de3,de7]).
db(pair,初期化JCLファイル,[]).
db(pair,復旧プログラムファイル,[]).
db(pair,工程管理復旧プログラムファイル,[]).
db(pair,端末制御プログラムファイル,[]).
db(pair,アクセス管理プログラムファイル,[]).
db(pair,系切離し制御ファイル,[di3,di7]).
db(pair,設計仕様履歴ファイル,[di3,di7]).
db(pair,復旧完了JCLファイル,[]).
db(pair,復旧プログラムファイル,[]).
db(pair,工程管理復旧プログラムファイル,[]).
db(pair,端末制御プログラムファイル,[]).
db(pair,アクセス管理プログラムファイル,[]).
```

```
tepsi916::>fd0>t3.tst

[db(disk,dal,生産管理サブシステム,live),db(disk,dal,生産管理サブシステム,die)].
[db(disk,dcl,生産管理サブシステム,live),db(disk,dcl,生産管理サブシステム,die)].
test(["FNAME","DISK"],[[工程レポートファイル,dal],[システム構成情報ファイル,dal],[コンソールログファイル,dal]]).
```

```
tepsi916::>fd0>c1.crule

error1(1,2000,rule1,[DISK,DSU],[2000]):-
      テストケース(テストケースの入出力,3,[DISK],[_,_,_,i]),
      操作(異なる,12,DSU,I).
```

```
teps1916::>fd0>e2.erule

error2(1, 2001, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2001]) :-
     テストケース(テストケースの入出力, 3, [FNAME, DISK],
                  [_, _, _, _, S, _, _]),
     操作(異なる, 12, STEP, S).
error2(1, 2002, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2002]) :-
     テストケース(テストケースの入出力, 3, [FNAME, DISK],
                  [_, _, _, _, _, M, _]),
     操作(異なる, 12, FREASON, M).
error2(1, 2003, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2003]) :-
     システム影響(使用可, 1, SYSINF),
     ファイル影響(重大, 1, VSOP).
error2(1, 2004, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2004]) :-
     システム影響(使用不可, 2, SYSINF),
     ファイル影響(軽微, 2, VSOP).
error2(1, 2005, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2005]) :-
     対ディスク状態(正常, 1, FNAME, DISK, _, LIVE),
     'TSS影響'(ファイル傷害影響あり, 21, INF).
error2(1, 2006, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2006]) :-
     対ディスク状態(正常, 1, FNAME, DISK, _, LIVE),
     システム影響(使用不可, 2, SYSINF).
error2(1, 2007, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2007]) :-
     対ディスク状態(正常, 1, FNAME, DISK, _, LIVE),
     ファイル影響(重大, 1, VSOP).
error2(1, 2008, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2008]) :-
     対ディスク状態(傷害, 2, FNAME, DISK, _, LIVE),
     'TSS影響'(ファイル傷害影響なし, 22, INF),
     システム影響(使用可, 1, SYSINF),
     ファイル影響(軽微, 2, VSOP).
error2(1, 2009, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2009]) :-
     対ディスク状態(傷害, 2, FNAME, DISK, _, LIVE),
     理由説明(ペア状態によりセットアップリカバリー不可, 31, FREASON).
error2(1, 2010, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2010]) :-
     対ディスク状態(傷害, 2, FNAME, DISK, _, LIVE),
     理由説明(ペア状態によりTSSに影響あり, 32, FREASON).
error2(1, 2011, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2011]) :-
     対ディスク状態(傷害, 2, FNAME, DISK, _, LIVE),
     理由説明(ペア状態によりシステム傷害, 33, FREASON).
error2(1, 2012, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2012]) :-
     not(ファイル種類(多重, 211, FNAME, DISK, KIND)),
     理由説明(多重ファイル, 211, FREASON).
error2(1, 2013, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2013]) :-
     ファイル種類(等しい, 99, FNAME, DISK, KIND),
     not(種別(メンバー, 2, KIND, ['2重', '3重'])),
     操作(メンバー, 2, FREASON, ['多重化ファイルのため',
          'ペアディスクも障害になるとセットアップリカバリーが不可能になるため',
          'ペアディスクも障害になるとTSS業務への影響があるため',
          'ペアディスクも障害になるとシステムが障害となるため'])).
```

```
error2(1, 2014, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2014]) :-
     ファイル特性(セットアップ不可, 11, FNAME, DISK, STATUS),
     not(理由説明(セットアップ不可, 11, FREASON)).
error2(1, 2015, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2015]) :-
     ファイル特性('セットアップ','ターミネーション不可', 12, FNAME, DISK, STATUS),
     not(理由説明(セットアップターミネーション不可, 12, FREASON)).
error2(1, 2016, rule2, [FNAME, DISK, INF, SYSINF,
          STEP, FREASON, VSOP], [2016]) :-
     ファイル特性('セットアップ','リカバリー不可', 13, FNAME, DISK, STATUS),
     not(理由説明(セットアップリカバリー不可, 13, FREASON)).
```

```
tepsi916::>fd0>c3.crule

error3(1,2017,rule3,[STEP,WHDAY,DOP],[2017]):-
      テストケース(日時,1,[X],[Y,Z]),
      操作(等しい,11,WHDAY,X),
      操作(等しい,11,STEP,Y),
      操作(異なる,12,DOP,Z).
error3(1,2018,rule3,[STEP,WHDAY,DOP],[2018]):-
      not(日時(平日時間内,11,WHDAY)),
      not(日時(土曜時間内,21,WHDAY)),
      手順展開(連結,11,DOP).
error3(1,2019,rule3,[STEP,WHDAY,DOP],[2019]):-
      not(日時(平日時間内,11,WHDAY)),
      not(日時(土曜時間内,21,WHDAY)),
      手順展開(連結し復元,12,DOP).
```

```
tepsi916::\fd0\assum.inf

def(top_def, [center_system]).
def(center_system, [situation, facility]).
def(situation, [disc, file]).
def(facility, [measure]).
def(disc, [disc_apply, disc_resurt]).
def(file, [file_apply, file_result]).
def(disc_apply, [d_use]).
def(disc_result, [d_influence, d_step, d_plan1, d_importance]).
def(file_apply, [name, task, sub_info, sort, pair]).
def(file_result, [f_influence, sys_influence, f_step1,
                  f_comment, f_supervise]).
def(measure, [f_step2, date, d_plan2]).
def(d_use, [use_1, use_2, use_3, use_4, use_5]).
def(d_influence, [d_inf1, d_inf2]).
def(d_step, [d_step1, d_step2, d_step3, d_step4, d_step5]).
def(d_plan1, [d_comment1, d_comment2, d_comment3, d_comment4]).
def(d_importance, [d_importance1, d_importance2,
                   d_importance3, d_importance4]).
def(name, [name1, name2]).
def(task, [task_job, task_sys]).
def(task_job, [job1, job2, job3, job4, job5]).
def(task_sys, [sys1, sys2, sys3]).
def(sub_info, [sub_disable, sub_require]).
def(sub_disable, [sub1, sub2, sub3]).
def(sub_require, [sub4, sub5, sub6]).
def(sort, [multi, special, unknown]).
def(multi, [sort1, sort2, sort3]).
def(special, [sysfile, workfile]).
def(sysfile, [sort4, sort6]).
def(workfile, [sort5, sort7]).
def(unknown, [sort8]).
def(pair, [pair_live, pair_die]).
def(f_influence, [f_inf1, f_inf2]).
def(sys_influence, [sys_inf1, sys_inf2]).
def(f_step1, [res_require, res_disable]).
def(res_require, [fstep1, fstep2, fstep3, fstep4, fstep5, fstep6, fstep7]).
def(res_disable, [fstep8, fstep9, fstep10, fstep11, fstep12]).
def(f_comment, [com_sub, com_sort, com_pair]).
def(com_sub, [com1, com2, com3]).
def(com_sort, [com_spec, com_multi]).
def(com_spec, [com5, com6]).
def(com_pair, [com8, com9, com10]).
def(f_supervise, [fsu1, fsu2]).
def(f_step2, [res_require2, res_disab12]).
def(res_require2, [fstep21, fstep22, fstep23, fstep24,
                   fstep25, fstep26, fstep27]).
def(res_disable2, [fstep28, fstep29, fstep210, fstep211, pfstep212]).
def(date, [date_in, date_out]).
def(date_in, [date1, date3]).
def(date_out, [date2, date4, date5]).
def(d_plan2, [d_comment21, d_comment22, d_comment23, d_comment24]).
```

```
tepsi916::>fd0>flag.flg

flag(p_group,[ディスク用途,ディスク用途,[all]]).
flag(p_group,[日時,日時,[all]]).
flag(p_group,['TSS影響','TSS影響',[all]]).
flag(p_group,[処置影響,処置影響,[all]]).
flag(p_group,[手順展開,手順展開,[all]]).
flag(p_group,[ディスク種類,ディスク種類,[all]]).
flag(p_group,[業務,業務,[all]]).
flag(p_group,[ファイル特性,ファイル特性,[all]]).
flag(p_group,[ファイル種類,ファイル種類,[all]]).
flag(p_group,[対ディスク状態,対ディスク状態,[all]]).
flag(p_group,[ファイル包含,ファイル包含,[all]]).
flag(p_group,[理由説明,理由説明,[all]]).
flag(p_group,[ファイル影響,ファイル影響,[all]]).
flag(p_group,[システム影響,システム影響,[all]]).
flag(strategy,[ディスク用途,2,[]]).
flag(strategy,[日時,2,[]]).
flag(strategy,['TSS影響',2,[]]).
flag(strategy,[処置影響,2,[]]).
flag(strategy,[手順展開,2,[]]).
flag(strategy,[ディスク種類,2,[]]).
flag(strategy,[業務,2,[]]).
flag(strategy,[ファイル特性,2,[]]).
flag(strategy,[ファイル種類,2,[]]).
flag(strategy,[対ディスク状態,2,[]]).
flag(strategy,[理由説明,2,[]]).
flag(strategy,[ファイル影響,2,[]]).
flag(strategy,[システム影響,2,[]]).
flag(ref,[use_1,[ディスク用途(生産管理,1,"DISK","K")]]).
flag(ref,[use_2,[ディスク用途(原価管理,2,"DISK","K")]]).
flag(ref,[use_3,[ディスク用途(本番系,3,"DISK","K")]]).
flag(ref,[use_4,[ディスク用途(待機系,4,"DISK","K")]]).
flag(ref,[use_5,[ディスク用途(予備バック,5,"DISK","K")]]).
flag(ref,[d_inf1,['TSS影響'(ディスク偏有影響あり,11,"INF")]]).
flag(ref,[d_inf2,['TSS影響'(ディスク偏有影響なし,12,"INF")]]).
flag(ref,[d_step1,[処置影響(ホットスタンバイ,11,"DSTEP")]]).
flag(ref,[d_step2,[処置影響(影響なし,12,"DSTEP")]]).
flag(ref,[d_step3,[処置影響(待機系ダウン,13,"DSTEP")]]).
flag(ref,[d_step4,[処置影響(業務使用中,14,"DSTEP")]]).
flag(ref,[d_step5,[処置影響(不明,15,"DSTEP")]]).
flag(ref,[d_comment1,[手順展開(連絡,11,"DOP")]]).
flag(ref,[d_comment2,[手順展開(連絡し復元,12,"DOP")]]).
flag(ref,[d_comment3,[手順展開(バックチェック,2,"DOP")]]).
flag(ref,[d_comment4,[手順展開(バックチェックし復元,3,"DOP")]]).
flag(ref,[d_importance1,[ディスク種類(本番系,1,"DSUP")]]).
flag(ref,[d_importance2,[ディスク種類(待機系,2,"DSUP")]]).
flag(ref,[d_importance3,[ディスク種類(予備,3,"DSUP")]]).
flag(ref,[d_importance4,[ディスク種類(その他,4,"DSUP")]]).
flag(ref,[name1,[ファイル包含(含む,1,"FNAME","FNAME2")]]).
flag(ref,[name2,[ファイル包含(含まない),2,"FNAME","FNAME2")]]).
flag(ref,[Job1,[業務(品質管理,11,"FNAME","DISK","TASK")]]).
flag(ref,[Job2,[業務(工程管理,12,"FNAME","DISK","TASK")]]).
flag(ref,[Job3,[業務(原価管理,13,"FNAME","DISK","TASK")]]).
flag(ref,[Job4,[業務(文書管理,14,"FNAME","DISK","TASK")]]).
flag(ref,[Job5,[業務(連絡,15,"FNAME","DISK","TASK")]]).
flag(ref,[sys1,[業務(共通1,211,"FNAME","DISK","TASK")]]).
flag(ref,[sys2,[業務(共通2,212,"FNAME","DISK","TASK")]]).
flag(ref,[sys3,[業務(課金,221,"FNAME","DISK","TASK")]]).
flag(ref,[sub1,
     [ファイル特性(セットアップ不可,11,"FNAME","DISK","STATUS")]]).
flag(ref,[sub2,
```

```
      [ファイル特性('セットアップ ターミネーション不可',12,"FNAME",
                "DISK","STATUS")]]),
flag(ref,[sub3,
      [ファイル特性('セットアップ/リカバリー不可',13,"FNAME",
                "DISK","STATUS")]]),
flag(ref,[sub4,
      [ファイル特性(バックリカバリー要,21,"FNAME","DISK","STATUS")]]),
flag(ref,[sub5,
      [ファイル特性('DVF切り換え要',22,"FNAME","DISK","STATUS")]]),
flag(ref,[sub6,
      [ファイル特性(その他,23,"FNAME","DISK","STATUS")]]),
flag(ref,[sort1,
      [ファイル種類('1重',11,"FNAME","DISK","KIND")]]),
flag(ref,[sort2,
      [ファイル種類('2重',12,"FNAME","DISK","KIND")]]),
flag(ref,[sort3,
      [ファイル種類('3重',13,"FNAME","DISK","KIND")]]),
flag(ref,[sort4,
      [ファイル種類(蓄積,211,"FNAME","DISK","KIND")]]),
flag(ref,[sort5,
      [ファイル種類(課金,221,"FNAME","DISK","KIND")]]),
flag(ref,[sort6,
      [ファイル種類(ジャーナル,212,"FNAME","DISK","KIND")]]),
flag(ref,[sort7,
      [ファイル種類(プログラム,222,"FNAME","DISK","KIND")]]),
flag(ref,[sort8,
      [ファイル種類(不明,3,"FNAME","DISK","KIND")]]),
flag(ref,[pair_live,
      [対ディスク状態(正常,1,"FNAME","DISK","PAIR","LIVE")]]),
flag(ref,[pair_die,
      [対ディスク状態(障害,2,"FNAME","DISK","PAIR","LIVE")]]),
flag(ref,[f_inf1,['TSS影響'(ファイル障害影響あり,21,"INF")]]),
flag(ref,[f_inf2,['TSS影響'(ファイル障害影響なし,22,"INF")]]),
flag(ref,[sys_inf1,[システム影響(使用可,1,"SYSINF")]]),
flag(ref,[sys_inf2,[システム影響(使用不可,2,"SYSINF")]]),
flag(ref,[fstep1,[処置影響(バック切換え,211,"STEP")]]),
flag(ref,[fstep2,[処置影響(ファイル復元,212,"STEP")]]),
flag(ref,[fstep3,[処置影響(バックリカバリー,213,"STEP")]]),
flag(ref,[fstep4,[処置影響('MTバックアップ',214,"STEP")]]),
flag(ref,[fstep5,[処置影響(再実行,215,"STEP")]]),
flag(ref,[fstep6,[処置影響(フォールバック,216,"STEP")]]),
flag(ref,[fstep7,[処置影響(処置不明,217,"STEP")]]),
flag(ref,[fstep8,[処置影響(スケジューリング不可,221,"STEP")]]),
flag(ref,[fstep9,[処置影響(業務不可,222,"STEP")]]),
flag(ref,[fstep10,[処置影響(文書管理不可,223,"STEP")]]),
flag(ref,[fstep11,[処置影響(セットアップリカバリー不可,224,"STEP")]]),
flag(ref,[fstep12,[処置影響(発注不可,225,"STEP")]]),
flag(ref,[com_spec,[理由説明(特殊ファイル,21,"REASON")]]),
flag(ref,[com1,[理由説明(セットアップ不可,11,"REASON")]]),
flag(ref,[com2,[理由説明(セットアップターミネーション不可,12,"REASON")]]),
flag(ref,[com3,[理由説明(セットアップリカバリー不可,13,"REASON")]]),
flag(ref,[com5,[理由説明(蓄積ファイル,211,"REASON")]]),
flag(ref,[com6,[理由説明(プログラム,212,"REASON")]]),
flag(ref,[com_multi,[理由説明(多重化ファイル,22,"REASON")]]),
flag(ref,[com8,[理由説明(ペア状態によりセットアップリカバリー不可,31,"REASON")]]),
flag(ref,[com9,[理由説明(ペア状態によりTSSに影響あり,32,"REASON")]]),
flag(ref,[com10,[理由説明(ペア状態によりシステム障害,33,"REASON")]]),
flag(ref,[fsu1,[ファイル影響(増大,1,"VSOP")]]),
flag(ref,[fsu2,[ファイル影響(解放,2,"VSOP")]]),
flag(ref,[fstep21,[処置影響(バック切換え,211,"STEP")]]),
flag(ref,[fstep22,[処置影響(ファイル復元,212,"STEP")]]),
flag(ref,[fstep23,[処置影響(バックリカバリー,213,"STEP")]]),
flag(ref,[fstep24,[処置影響('MTバックアップ',214,"STEP")]]),
flag(ref,[fstep25,[処置影響(再実行,215,"STEP")]]),
flag(ref,[fstep26,[処置影響(フォールバック,216,"STEP")]]),
flag(ref,[fstep27,[処置影響(処置不明,217,"STEP")]]),
```

```
flag(ref,[[step28,[処置影響(スケジューリング不可,221,"STEP")]],
flag(ref,[[step29,[処置影響(業務不可,223,"STEP")]]),
flag(ref,[[fstep210,[処置影響(文書管理不可,223,"STEP")]]),
flag(ref,[[fstep211,[処置影響(セットアップリカバリー不可,224,"STEP")]]]),
flag(ref,[[fstep212,[処置影響(発注不可,225,"STEP")]]]),
flag(ref,[date1,[日時(平日時間内,11,"WHDAY")]]),
flag(ref,[date2,[日時(平日時間外,12,"WHDAY")]]),
flag(ref,[date3,[日時(土曜時間内,21,"WHDAY")]]),
flag(ref,[date4,[日時(土曜時間外,22,"WHDAY")]]),
flag(ref,[date5,[日時(日曜,3,"WHDAY")]]]),
flag(ref,[d_comment21,[手順展開(連絡,11,"DOP")]]),
flag(ref,[d_comment22,[手順展開(連絡し復元,12,"DOP")]]),
flag(ref,[d_comment23,[手順展開(バックチェック,2,"DOP")]]),
flag(ref,[d_comment24,[手順展開(バックチェックし復元,3,"DOP")]]]).
flag(error,[2002,[ファイル種類,ファイル特性,対ディスク状態,処置影響,
                  'TSS影響',システム影響,理由説明]]).
flag(error,[2004,[ファイル種類,ファイル特性,'TSS影響',システム影響]]).
flag(error,[2005,[ファイル種類,ファイル特性,対ディスク状態,'TSS影響',
                  システム影響]]).
flag(error,[2008,[ファイル種類,ファイル特性,対ディスク状態,'TSS影響',
                  システム影響,ファイル影響]]).
```

B.3　検証結果

検証過程と検証結果に関する以下の表示画面を次ページ以降に示す。

1)対話ウィンド

検証の開始から終了までの対話画面

生成仮説と成功仮説（生成仮説の検証結果）の表示

2)推論ネットワーク

ＡＴＭＳノードの表示

3)グラフィック表示例

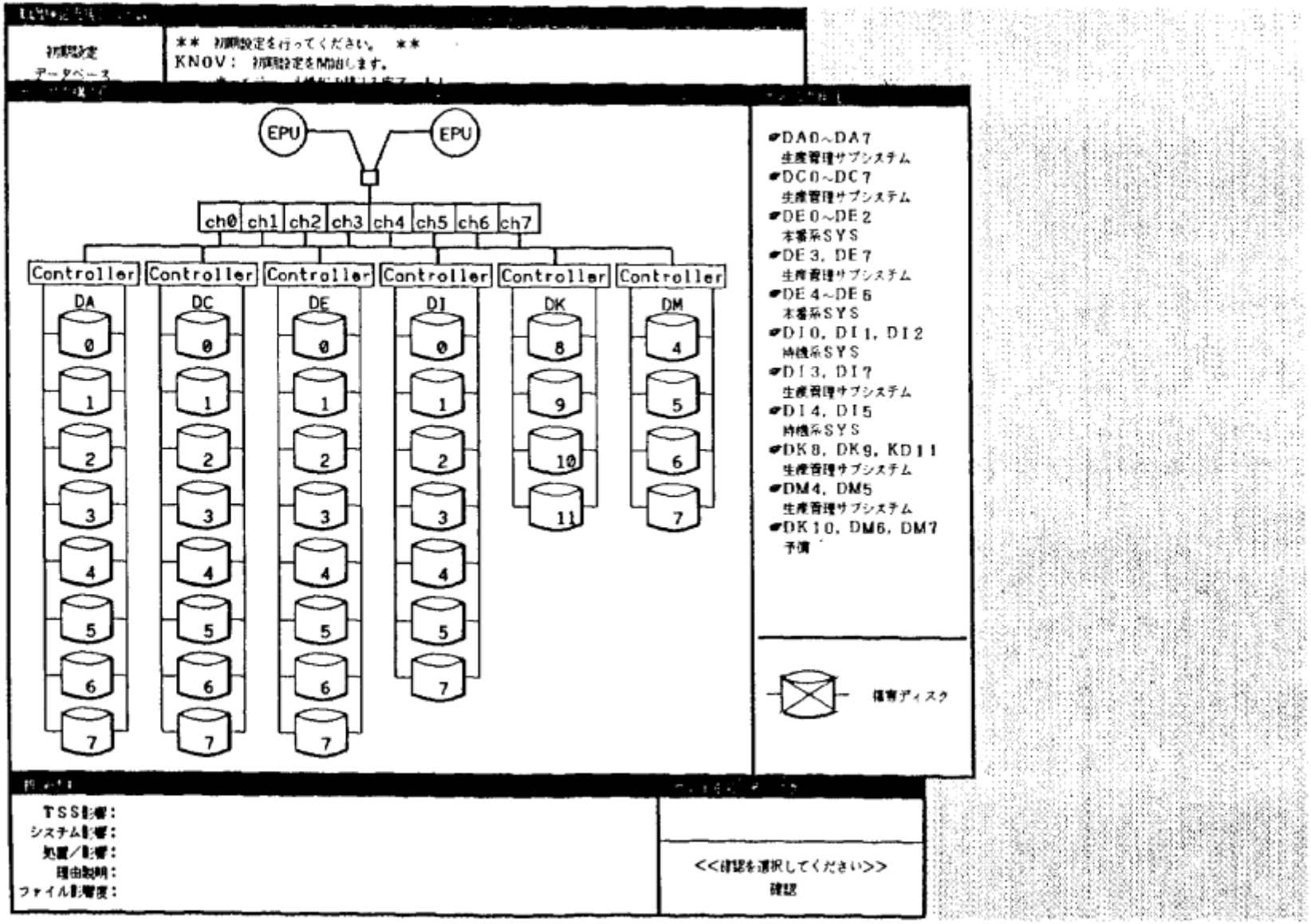
補助情報表示クラスとして"cg"を指定した場合の表示。

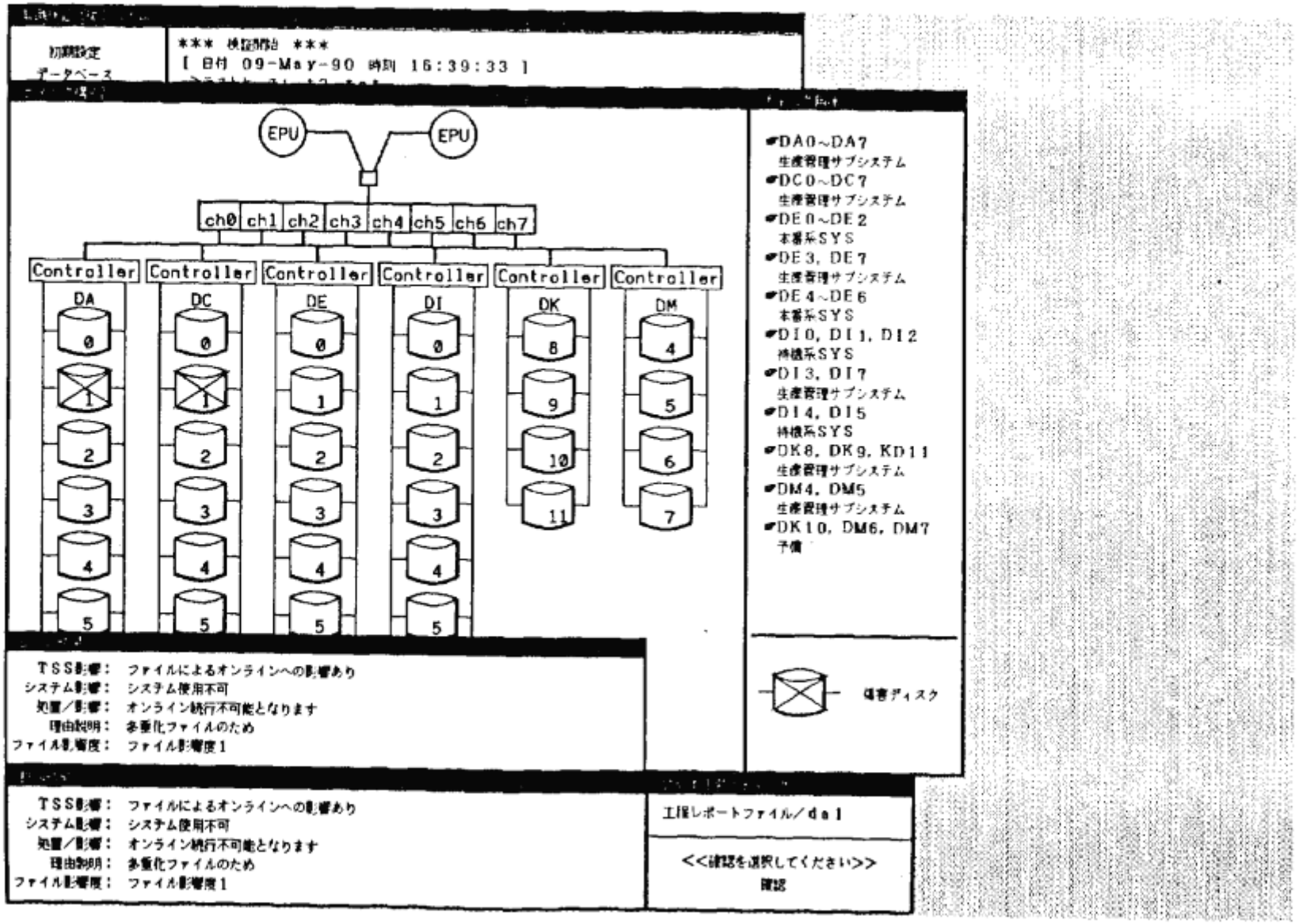
この場合にはクラス"cg"を予めカタログしておく必要がある。

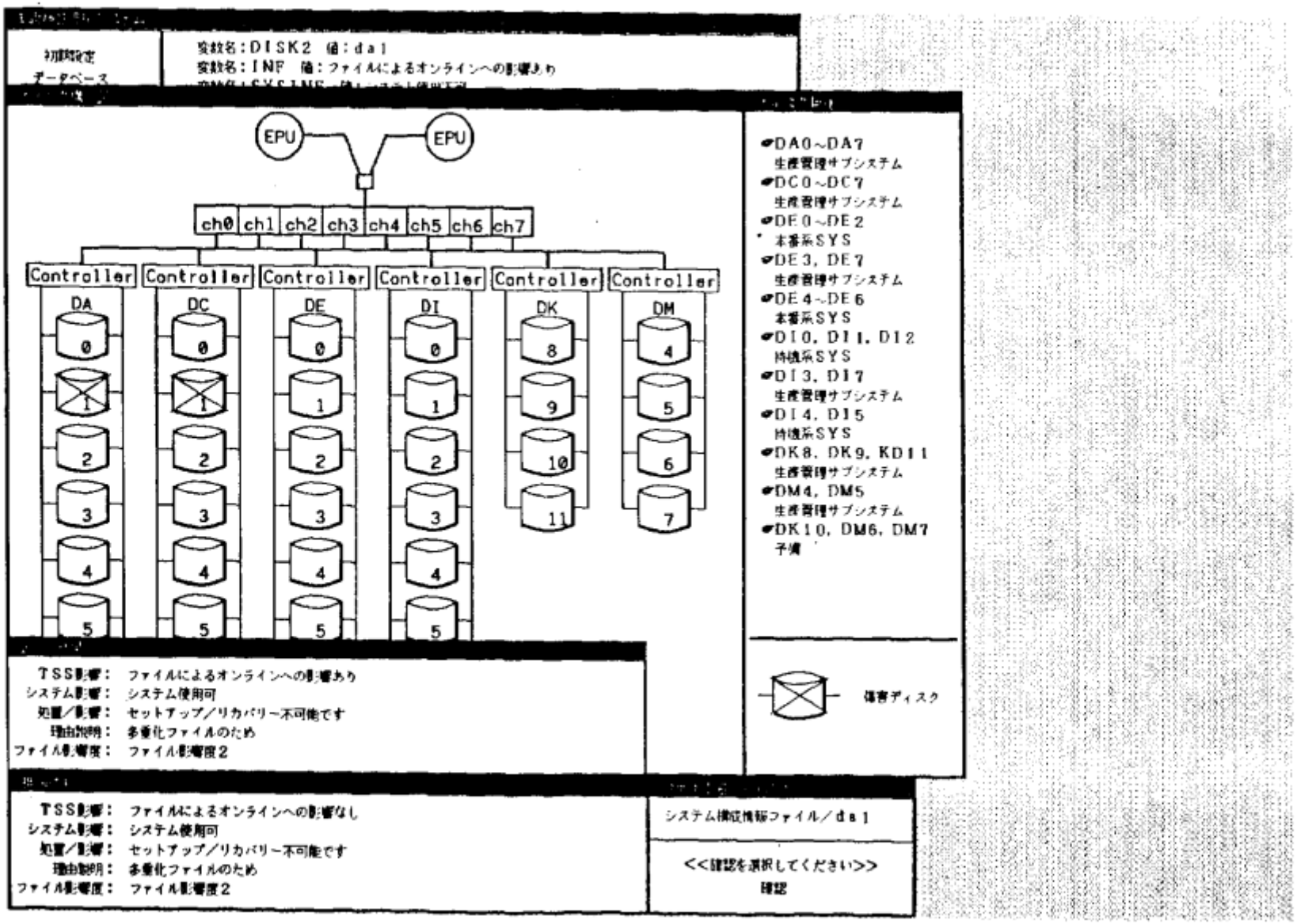
```
＊＊＊　検証開始　＊＊＊
［　時刻　16：37：31　］
　＞テストケース：　t3．tst
　＞推論方式：　全探索　　＞トレース：　off
------------------------------------------------------------
＊＊＊　＜　rule2　＞
｜入力値：［工程レポートファイル，da1］
－＞優先度＝1　ID＝19　succeed．
　　変数名：FNAME1　値：工程レポートファイル
　　変数名：DISK2　値：da1
　　変数名：INF　値：ファイルによるオンラインへの影響あり
　　変数名：SYSINF　値：システム使用不可
　　変数名：STEP　値：オンライン続行不可能となります
　　変数名：FREASON　値：多重化ファイルのため
　　変数名：VSOP　値：ファイル影響度1
＊＊＊　＜　終了　＞　＊＊＊
＊＊＊　＜　rule2　＞
｜入力値：［システム構成情報ファイル，da1］
－＞優先度＝1　ID＝10　succeed．
　　変数名：FNAME1　値：システム構成情報ファイル
　　変数名：DISK2　値：da1
　　変数名：INF　値：ファイルによるオンラインへの影響なし
　　変数名：SYSINF　値：システム使用可
　　変数名：STEP　値：セットアップ／リカバリー不可能です
　　変数名：FREASON　値：多重化ファイルのため
　　変数名：VSOP　値：ファイル影響度2
＊＊＊　＜　終了　＞　＊＊＊
＊＊＊　＜　rule2　＞
｜入力値：［コンソールログファイル，da1］
－＞優先度＝1　ID＝18　succeed．
　　変数名：FNAME1　値：コンソールログファイル
　　変数名：DISK2　値：da1
　　変数名：INF　値：ファイルによるオンラインへの影響あり
　　変数名：SYSINF　値：システム使用不可
　　変数名：STEP　値：オンライン続行不可能となります
　　変数名：FREASON　値：多重化ファイルのため
　　変数名：VSOP　値：ファイル影響度1
＊＊＊　＜　終了　＞　＊＊＊
∞∞∞　［　error2　］　矛盾検出ルール実行
∞∞∞　［　終了　！　∞∞∞
＊＊＊　矛盾検出により、次のルールが矛盾です．
　rule2（1，10，［A，B］，［A，B，C，D，E，F，G］）：－
　　ファイル包含（含む，1，A，［システム構成情報ファイル］），
　　対ディスク状態（障害，2，A，B，H，I），
　　TSS影響（ファイル障害影響なし，22，C），
　　システム影響（使用可，1，D），
　　処置影響（セットアップリカバリー不可，224，E），
　　理由説明（多重化ファイル，22，F），
　　ファイル影響（軽微，2，G），
仮説生成を行いますか　（y，／n，）　？－y．

＊＊＊　仮説生成終了　［1］　＊＊＊

------------------------------------------------------------
［　時刻　16：37：54　］
＊＊＊　検証終了　＊＊＊

＊＊＊　仮説表示開始　＊＊＊
ルール名：rule2　　開始ID：　25
全仮説数：1　　無矛盾仮説数　1　　矛盾仮説数－1
成功仮説数　0　　成功仮説数　1　　成功仮説数（唯一結果）　0
```

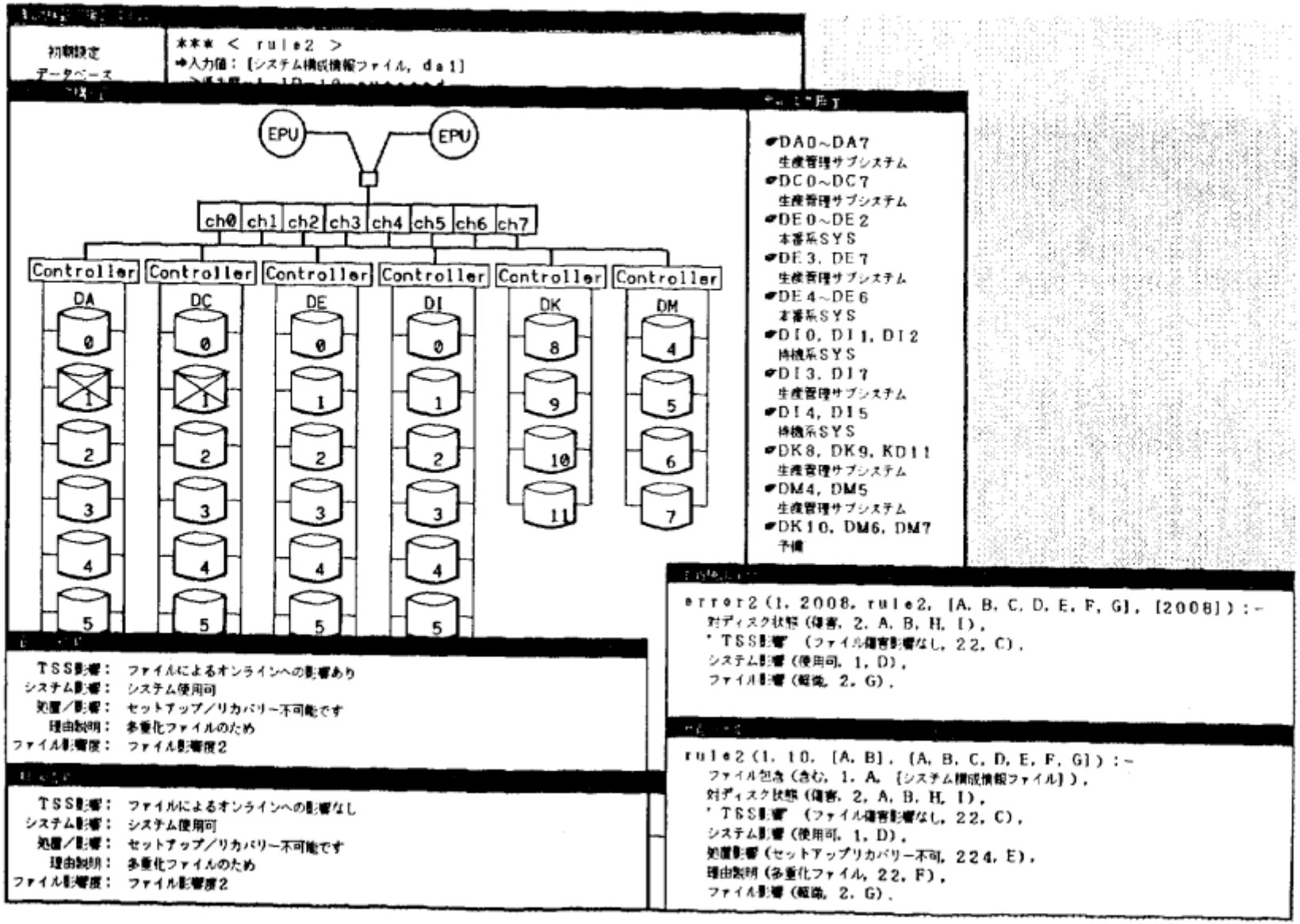
```
ＩＤ表示　（ｉｄ．）　仮説表示　（ａｄ．）　終了　（ｅ．）
選択＞ｉｄ．
１．無矛盾仮説　２．矛盾仮説　３．未成功仮説　４．成功仮説　５．成功仮説（同一結果）
番号＞４．
【　成功仮説　】
　◆　２７　◆　２８　◆　３１
番号＞ｅ．
ＩＤ表示　（ｉｄ．）　仮説表示　（ａｄ．）　終了　（ｅ．）
選択＞ａｄ．
表示する仮説ＩＤを入力してください．　終了（ｅ．）
ＩＤ＞２７．
□ＩＤ＝２７　　仮説　　状態：無矛盾
　ｒｕｌｅ２（１，２７，［Ａ，Ｂ］，［Ａ，Ｂ，Ｃ，Ｄ，Ｅ，Ｆ，Ｇ］）：－
　　ファイル包含（含む，１，Ａ，［システム構成情報ファイル］），
　　対ディスク状態（傷害，２，Ａ，Ｂ，Ｈ，Ｉ），
　　ＴＳＳ影響（ファイル傷害影響なし，２２，Ｃ），
　　システム影響（使用不可，２，Ｄ），
　　処置影響（セットアップリカバリー不可，２２４，Ｅ），
　　理由説明（多重化ファイル，２２，Ｆ），
　　ファイル影響（重大，１，Ｇ）．
ＩＤ＞２８．
□ＩＤ＝２８　　仮説　　状態：無矛盾
　ｒｕｌｅ２（１，２８，［Ａ，Ｂ］，［Ａ，Ｂ，Ｃ，Ｄ，Ｅ，Ｆ，Ｇ］）：－
　　ファイル包含（含む，１，Ａ，［システム構成情報ファイル］），
　　対ディスク状態（傷害，２，Ａ，Ｂ，Ｈ，Ｉ），
　　ＴＳＳ影響（ファイル傷害影響あり，２１，Ｃ），
　　システム影響（使用可，１，Ｄ），
　　処置影響（セットアップリカバリー不可，２２４，Ｅ），
　　理由説明（多重化ファイル，２２，Ｆ），
　　ファイル影響（軽微，２，Ｇ）．
ＩＤ＞３１．
□ＩＤ＝３１　　仮説　　状態：無矛盾
　ｒｕｌｅ２（１，３１，［Ａ，Ｂ］，［Ａ，Ｂ，Ｃ，Ｄ，Ｅ，Ｆ，Ｇ］）：－
　　ファイル包含（含む，１，Ａ，［システム構成情報ファイル］），
　　対ディスク状態（傷害，２，Ａ，Ｂ，Ｈ，Ｉ），
　　ＴＳＳ影響（ファイル傷害影響あり，２１，Ｃ），
　　システム影響（使用不可，２，Ｄ），
　　処置影響（セットアップリカバリー不可，２２４，Ｅ），
　　理由説明（多重化ファイル，２２，Ｆ），
　　ファイル影響（重大，１，Ｇ）．
```

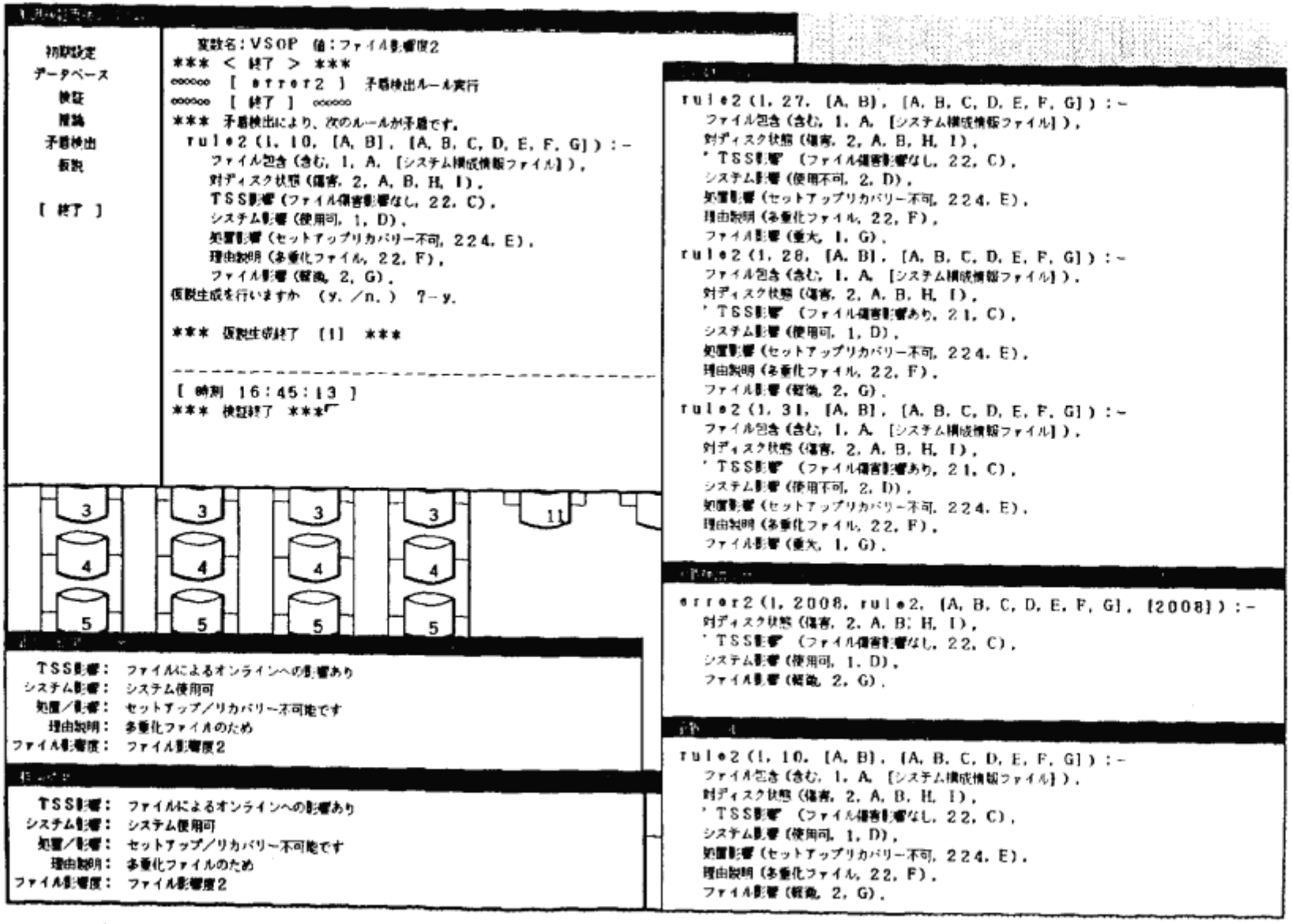
```
tepsi916::2fd02fc33.log

***  インスタンス・ルールノード  表示開始  ***
□ルール名：rule2  引数：2  ノード数：10    ○ 無矛盾    ⊥ 矛盾
○  node#1  導出ルールID：[19]
       入力値：[工程レポートファイル，dal]
       出力値：[工程レポートファイル，dal，ファイルによるオンラインへの影響あり，システム使用不可，オンライン続行不可能となります，多重化ファイルのため，ファイル影響度1]

○  node#2  導出ルールID：[18]
       入力値：[コンソールログファイル，dal]
       出力値：[コンソールログファイル，dal，ファイルによるオンラインへの影響あり，システム使用不可，オンライン続行不可能となります，多重化ファイルのため，ファイル影響度1
]

⊥  node#3  導出ルールID：[10]
       入力値：[システム構成情報ファイル，dal]
       出力値：[システム構成情報ファイル，dal，ファイルによるオンラインへの影響なし，システム使用可，セットアップ／リカバリー不可能です，多重化ファイルのため，ファイル影響
度2
       ]
⊥  node#4  導出ルールID：[25]
       入力値：[システム構成情報ファイル，dal]
       出力値：[システム構成情報ファイル，dal，ファイルによるオンラインへの影響なし，システム使用可，セットアップ／リカバリー不可能です，多重化ファイルのため，ファイル影響
度1
       ]
⊥  node#5  導出ルールID：[26]
       入力値：[システム構成情報ファイル，dal]
       出力値：[システム構成情報ファイル，dal，ファイルによるオンラインへの影響なし，システム使用不可，セットアップ／リカバリー不可能です，多重化ファイルのため，ファイル影
響度
       2]
○  node#6  導出ルールID：[27]
       入力値：[システム構成情報ファイル，dal]
       出力値：[システム構成情報ファイル，dal，ファイルによるオンラインへの影響なし，システム使用不可，セットアップ／リカバリー不可能です，多重化ファイルのため，ファイル影
響度
       1]
○  node#7  導出ルールID：[28]
       入力値：[システム構成情報ファイル，dal]
       出力値：[システム構成情報ファイル，dal，ファイルによるオンラインへの影響あり，システム使用可，セットアップ／リカバリー不可能です，多重化ファイルのため，ファイル影響
度2
       ]
⊥  node#8  導出ルールID：[29]
       入力値：[システム構成情報ファイル，dal]
       出力値：[システム構成情報ファイル，dal，ファイルによるオンラインへの影響あり，システム使用可，セットアップ／リカバリー不可能です，多重化ファイルのため，ファイル影響
度1
       ]
⊥  node#9  導出ルールID：[30]
       入力値：[システム構成情報ファイル，dal]
       出力値：[システム構成情報ファイル，dal，ファイルによるオンラインへの影響あり，システム使用不可，セットアップ／リカバリー不可能です，多重化ファイルのため，ファイル影
響度
       2]
○  node#10  導出ルールID：[31]
       入力値：[システム構成情報ファイル，dal]
       出力値：[システム構成情報ファイル，dal，ファイルによるオンラインへの影響あり，システム使用不可，セットアップ／リカバリー不可能です，多重化ファイルのため，ファイル影
響度
       1]
***  表示終了  ***
```


初期設定
データベース
** 初期設定を行ってください。 **
KNOV: 初期設定を開始します。
EPU
EPU
ch0 ch1 ch2 ch3 ch4 ch5 ch6 ch7
Controller
Controller
Controller
Controller
Controller
Controller
DA
DC
DE
DI
DK
DM
●DA0～DA7
生産管理サブシステム
●DC0～DC7
生産管理サブシステム
●DE0～DE2
本番系SYS
●DE3, DE7
生産管理サブシステム
●DE4～DE6
本番系SYS
●DI0, DI1, DI2
待機系SYS
●DI3, DI7
生産管理サブシステム
●DI4, DI5
待機系SYS
●DK8, DK9, KD11
生産管理サブシステム
●DM4, DM5
生産管理サブシステム
●DK10, DM6, DM7
予備
保守ディスク
TSS影響:
システム影響:
処置/影響:
理由説明:
ファイル影響度:
<<確認を選択してください>>
確認


初期設定
データベース
*** 検証開始 ***
[日付 09-May-90 時刻 16:39:33]
EPU
EPU
ch0 ch1 ch2 ch3 ch4 ch5 ch6 ch7
Controller
Controller
Controller
Controller
Controller
Controller
DA
DC
DE
DI
DK
DM
●DA0～DA7
生産管理サブシステム
●DC0～DC7
生産管理サブシステム
●DE0～DE2
本番系SYS
●DE3, DE7
生産管理サブシステム
●DE4～DE6
本番系SYS
●DI0, DI1, DI2
待機系SYS
●DI3, DI7
生産管理サブシステム
●DI4, DI5
待機系SYS
●DK8, DK9, KD11
生産管理サブシステム
●DM4, DM5
生産管理サブシステム
●DK10, DM6, DM7
予備
TSS影響： ファイルによるオンラインへの影響あり
システム影響： システム使用不可
処置／影響： オンライン続行不可能となります
理由説明： 多重化ファイルのため
ファイル影響度： ファイル影響度1
障害ディスク
TSS影響： ファイルによるオンラインへの影響あり
システム影響： システム使用不可
処置／影響： オンライン続行不可能となります
理由説明： 多重化ファイルのため
ファイル影響度： ファイル影響度1
工程レポートファイル／da1
<<確認を選択してください>>
確認


初期設定
データベース
変数名：DISK2 値：da1
変数名：INF 値：ファイルによるオンラインへの影響あり
EPU
EPU
ch0 ch1 ch2 ch3 ch4 ch5 ch6 ch7
Controller
Controller
Controller
Controller
Controller
Controller
DA
DC
DE
DI
DK
DM
●DA0～DA7
生産管理サブシステム
●DC0～DC7
生産管理サブシステム
●DE0～DE2
本番系SYS
●DE3，DE7
生産管理サブシステム
●DE4～DE6
本番系SYS
●DI0，DI1，DI2
待機系SYS
●DI3，DI7
生産管理サブシステム
●DI4，DI5
待機系SYS
●DK8，DK9，KD11
生産管理サブシステム
●DM4，DM5
生産管理サブシステム
●DK10，DM6，DM7
予備
TSS影響： ファイルによるオンラインへの影響あり
システム影響： システム使用可
処置/影響： セットアップ/リカバリー不可能です
理由説明： 多重化ファイルのため
ファイル影響度： ファイル影響度2
障害ディスク
TSS影響： ファイルによるオンラインへの影響なし
システム影響： システム使用可
処置/影響： セットアップ/リカバリー不可能です
理由説明： 多重化ファイルのため
ファイル影響度： ファイル影響度2
システム構成情報ファイル/da1
<<確認を選択してください>>
確認


初期設定
データベース
*** < rule2 >
➡入力値：[システム構成情報ファイル, da1]
EPU
EPU
ch0 ch1 ch2 ch3 ch4 ch5 ch6 ch7
Controller
Controller
Controller
Controller
Controller
Controller
DA
DC
DE
DI
DK
DM
●DA0～DA7
生産管理サブシステム
●DC0～DC7
生産管理サブシステム
●DE0～DE2
本番系SYS
●DE3, DE7
生産管理サブシステム
●DE4～DE6
本番系SYS
●DI0, DI1, DI2
待機系SYS
●DI3, DI7
生産管理サブシステム
●DI4, DI5
待機系SYS
●DK8, DK9, KD11
生産管理サブシステム
●DM4, DM5
生産管理サブシステム
●DK10, DM6, DM7
予備
TSS影響： ファイルによるオンラインへの影響あり
システム影響： システム使用可
処置／影響： セットアップ／リカバリー不可能です
理由説明： 多重化ファイルのため
ファイル影響度： ファイル影響度2
TSS影響： ファイルによるオンラインへの影響なし
システム影響： システム使用可
処置／影響： セットアップ／リカバリー不可能です
理由説明： 多重化ファイルのため
ファイル影響度： ファイル影響度2
error2(1, 2008, rule2, [A, B, C, D, E, F, G], [2008]) :-
対ディスク状態(障害, 2, A, B, H, I),
' TSS影響 (ファイル障害影響なし, 22, C),
システム影響(使用可, 1, D),
ファイル影響(軽微, 2, G).
rule2(1, 10, [A, B], [A, B, C, D, E, F, G]) :-
ファイル包含(含む, 1, A, [システム構成情報ファイル]),
対ディスク状態(障害, 2, A, B, H, I),
' TSS影響 (ファイル障害影響なし, 22, C),
システム影響(使用可, 1, D),
処置影響(セットアップリカバリー不可, 224, E),
理由説明(多重化ファイル, 22, F),
ファイル影響(軽微, 2, G).


初期設定
データベース
検証
推論
予盾検出
仮説
[終了]
変数名：VSOP 値：ファイル影響度2
*** < 終了 > ***
∞∞∞ [error2] 予盾検出ルール実行
∞∞∞ [終了] ∞∞∞
*** 予盾検出により、次のルールが予盾です。
rule2(1, 10, [A, B], [A, B, C, D, E, F, G]) :-
ファイル包含(含む, 1, A, [システム構成情報ファイル]),
対ディスク状態(障害, 2, A, B, H, I),
TSS影響(ファイル障害影響なし, 22, C),
システム影響(使用可, 1, D),
処置影響(セットアップリカバリー不可, 224, E),
理由説明(多重化ファイル, 22, F),
ファイル影響(軽微, 2, G).
仮説生成を行いますか (y. /n.) ?-y.
*** 仮説生成終了 [1] ***
[時刻 16:45:13]
*** 検証終了 ***
rule2(1, 27, [A, B], [A, B, C, D, E, F, G]) :-
ファイル包含(含む, 1, A, [システム構成情報ファイル]),
対ディスク状態(障害, 2, A, B, H, I),
' TSS影響 (ファイル障害影響なし, 22, C),
システム影響(使用可, 2, D),
処置影響(セットアップリカバリー不可, 224, E),
理由説明(多重化ファイル, 22, F),
ファイル影響(重大, 1, G).
rule2(1, 28, [A, B], [A, B, C, D, E, F, G]) :-
ファイル包含(含む, 1, A, [システム構成情報ファイル]),
対ディスク状態(障害, 2, A, B, H, I),
' TSS影響 (ファイル障害影響あり, 21, C),
システム影響(使用可, 1, D),
処置影響(セットアップリカバリー不可, 224, E),
理由説明(多重化ファイル, 22, F),
ファイル影響(軽微, 2, G).
rule2(1, 31, [A, B], [A, B, C, D, E, F, G]) :-
ファイル包含(含む, 1, A, [システム構成情報ファイル]),
対ディスク状態(障害, 2, A, B, H, I),
' TSS影響 (ファイル障害影響あり, 21, C),
システム影響(使用不可, 2, D),
処置影響(セットアップリカバリー不可, 224, E),
理由説明(多重化ファイル, 22, F),
ファイル影響(重大, 1, G).
error2(1, 2008, rule2, [A, B, C, D, E, F, G], [2008]) :-
対ディスク状態(障害, 2, A, B, H, I),
' TSS影響 (ファイル障害影響なし, 22, C),
システム影響(使用可, 1, D),
ファイル影響(軽微, 2, G).
rule2(1, 10, [A, B], [A, B, C, D, E, F, G]) :-
ファイル包含(含む, 1, A, [システム構成情報ファイル]),
対ディスク状態(障害, 2, A, B, H, I),
' TSS影響 (ファイル障害影響なし, 22, C),
システム影響(使用可, 1, D),
処置影響(セットアップリカバリー不可, 224, E),
理由説明(多重化ファイル, 22, F),
ファイル影響(軽微, 2, G).
3
4
5
11
TSS影響： ファイルによるオンラインへの影響あり
システム影響： システム使用可
処置／影響： セットアップ／リカバリー不可能です
理由説明： 多重化ファイルのため
ファイル影響度： ファイル影響度2
TSS影響： ファイルによるオンラインへの影響あり
システム影響： システム使用可
処置／影響： セットアップ／リカバリー不可能です
理由説明： 多重化ファイルのため
ファイル影響度： ファイル影響度2