

J A E R I - M

88-152

原子力コードのベクトル化支援
エキスパート・システムCes-VP

1988年8月

藤崎 正英*・牧野 光弘*・石黒美佐子

日本原子力研究所
Japan Atomic Energy Research Institute

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

JAERI-M reports are issued irregularly.

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

©Japan Atomic Energy Research Institute, 1988

編集兼発行 日本原子力研究所
印 刷 いばらき印刷株

原子力コードのベクトル化支援エキスパート・システム Ces-VP

日本原子力研究所東海研究所計算センター

藤崎正英^{*}・牧野光弘^{*}・石黒美佐子

(1988年7月20日受理)

10年間にわたる日本原子力研究所でのベクトル化経験を通じて蓄積されたベクトル化ノウハウをルールとしてまとめた原子力コードのベクトル化支援を行うエキスパート・システム Ces-VP (Consultation Expert System for Vector Programming of nuclear codes)を試作した。ベクトル化経験を豊富にもつ専門家が、初心者に比べて短時間で高性能のコードに再構成することができる。これは、専門家は、いくつもの原子力コードをベクトル化してきた経験から多くのベクトル化技法の中から最適なものを選択して修正するノウハウを多く持っているからである。そこで、ベクトル化の初心者がベクトル化ノウハウを手軽に引き出せるようなシステムがあれば、ベクトル化技法の選択に要する試行錯誤を減少させることができる。本報告書では、はじめに Ces-VP の内容とその開発目的について述べる。現在までのベクトル化ノウハウの獲得とルール化するための方法についても述べる。そして富士通エキスパート・システム構築ツール ESHELL を用いて作成した Ces-VP の概要を紹介する。さらに Ces-VP を実際に利用して得たアンケートの結果を示し、この結果を基に Ces-VP の評価及び問題点の考察を行う。

Ces-VP : Consultation Expert System for Vector Programming
of Nuclear Codes

Masahide FUJISAKI*, Mitsuhiro MAKINO* and Misako ISHIGURO

Computing Center
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received July 20, 1988)

Ces-VP is a prototype rule-based expert system for consulting the vector programming, based on the knowledge of vectorization of nuclear codes at JAERI during these 10 years. Experts in vectorization can restructure nuclear codes with high performance on vector processors, since they have know-how for choosing the best technique among a lot of techniques that were acquired from the experience of vectorization in the past. Frequency in trial and error will be reduced if a beginner can easily use the know-how of experts. In this report, at first the contents of Ces-VP and its intention are shown. Then, the method for acquiring the know-how of vectorization and the method for making rules from the know-how are described. The outline of Ces-VP implemented on Fujitsu expert tool ESHELL is described. Finally, the availability of Ces-VP is evaluated from the data gathered from practical use and its present problems are discussed.

Keywords : Vectorization, Rule Base, Expert Systems, Artificial Intelligence, Supercomputers

* On leave from FUJITSU LIMITED

目 次

1. はじめに	1
2. 原子力コードのベクトル化知識のルール化	3
3. Ces-VP システムの概要	18
3.1 機能概要	18
3.2 エキスパート・システム構築ツール	18
3.3 Ces-VP の構成	21
3.4 実行例	24
4. Ces-VP の評価と議論	34
4.1 アンケート要領	34
4.2 アンケート結果	34
4.3 議論	36
5. おわりに	74
謝辞	74
参考文献	75
付録 知識ベースのルール	76

Contents

1. Introduction	1
2. Building rules for vectorization of nuclear codes	3
3. The outline of Ces-VP system	18
3.1 The outline of functions	18
3.2 The tool for building an expert system	18
3.3 System configuration of Ces-VP	21
3.4 Examples of execution	24
4. Evaluation and discussion of Ces-VP system	34
4.1 The method of questionnaire	34
4.2 Analysis of results	34
4.3 Discussions	36
5. Concluding remarks	74
Acknowledgement	74
References	75
Appendix Rules of knowledge base	76

List of Tables

- Table 2.1 Vectorization technique recommended by Ces-VP and selection parameters
Table 2.2 Reasoning table for abend code OC6
Table 4.1 Optimal technique for problems of questionnaire
Table 4.2 Summary of results of questionnaire
Table 4.3 The summary of answer for problem no.1
Table 4.4 The summary of answer for problem no.2
Table 4.5 The summary of answer for problem no.3
Table 4.6 The summary of answer for problem no.4
Table 4.7 The summary of answer for problem no.5
Table 4.8 The summary of answer for problem no.6
Table 4.9 The summary of answer for problem no.7
Table 4.10 The summary of answer for problem no.8
Table 4.11 The summary of answer for problem no.9
Table 4.12 The summary of answer for problem no.10
Table 4.13 The summary of answer for problem no.11
Table 4.14 The summary of answer for problem no.12
Table 4.15 The summary of answer for problem no.13
Table 4.16 The summary of answer for problem no.14

List of Figures

- Fig. 2.1 Vectorization by transfer of DO loop between procedures.
(a) original program
(b) vectorized program
- Fig. 2.2 Vectorization by specifying the relationship of the size subscript of array element.
(a) original program
(b) vectorized program
- Fig. 2.3 Vectorization by changing the order of DO loops for removal of recursive reference.
(a) original program
(b) vectorized program
- Fig. 2.4 Vectorization by separating the DO loop at the I/O statements.
(a) original program
(b) vectorized program
- Fig. 2.5 Vector optimization by specifying vector length.
(a) original program
(b) vectorized program
- Fig. 2.6 An example of abend code OC6 by double precision data not no 8-byte boundary.
- Fig. 2.7 An example of abend code OC6 by specifying the incorrect number of REPEAT.
- Fig. 2.8 An example of abend code OC7 by using the nonnormalized data.
- Fig. 3.1 System configuration of expert shell ESHELL.
- Fig. 3.2 The relation between the components of ESHELL.
- Fig. 3.3 The order of rule search.
(a) depth-first-search
(b) width-first-search
- Fig. 3.4 System configuration of Ces-VP.
- Fig. 3.5 Knowledge source structure of Ces-VP.
- Fig. 3.6 An example of vectorizing program in Fig. 2.1 by using Ces-VP.
- Fig. 3.7 An example of fixing program in Fig. 2.6 by using Ces-VP.
- Fig. 3.8 An example of fixing program in Fig. 2.8 by using Ces-VP.
- Fig. 4.1 (a) Original source program of problem no.1
(b) Vectorized program of problem no.1 by expert
- Fig. 4.2 (a) Original source program of problem no.2
(b) Vectorized program of problem no.2 by expert

- Fig. 4.3 (a) Original source program of problem no.3
(b) Vectorized program of problem no.3 by expert
- Fig. 4.4 (a) Original source program of problem no.4
(b) Vectorized program of problem no.4 by expert
- Fig. 4.5 (a) Original source program of problem no.5
(b) Vectorized program of problem no.5 by expert
- Fig. 4.6 (a) Original source program of problem no.6
(b) Vectorized program of problem no.6 by expert
- Fig. 4.7 (a) Original source program of problem no.7
(b) Vectorized program of problem no.7 by expert
- Fig. 4.8 (a) Original source program of problem no.8
(b) Vectorized program of problem no.8 by expert
- Fig. 4.9 (a) Original source program of problem no.9
(b) Vectorized program of problem no.9 by expert
- Fig. 4.10 (a) Original source program of problem no.10
(b) Vectorized program of problem no.10 by expert
- Fig. 4.11 (a) Original source program of problem no.11
(b) Vectorized program of problem no.11 by expert
- Fig. 4.12 (a) Original source program of problem no.12
(b) Vectorized program of problem no.12 by expert
- Fig. 4.13 (a) Original source program of problem no.13
(b) Vectorized program of problem no.13 by expert
- Fig. 4.14 (a) Original source program of problem no.14
(b) Vectorized program of problem no.14 by expert

1. はじめに

原子力分野における計算プログラム（以下原子力コードと呼ぶ）のベクトル化を支援するエキスパート・システム Ces-VP (Consultation Expert System for Vector Programming of nuclear codes)を試作した。

原子力コードは、大量で複雑なシミュレーション計算が必要であり、計算時間は最高速の逐次処理計算機を使用しても1ケースに数分から数百時間もかかる。したがって、計算時間の短縮は必要不可欠である。そのため日本原子力研究所（以下原研と呼ぶ）では約10年前から原子力コードのベクトル処理計算機（以下ベクトル計算機と呼ぶ）による並列処理への適応性を調査し、ベクトル計算機の利用技術の研究を行ってきた。¹⁻⁷⁾その研究結果の一つの成果として、既存の原子力コードは元来ベクトル計算機の利用を意識して作られたものが少なく、プログラム構造やデータ構造に特徴があるため現状のベクトル・コンパイラの自動ベクトル化機能では十分ベクトル化されないことがわかった。そこで、ベクトル計算機の能力を引き出すために専門家によるベクトル化およびチューニングというプログラムの再構成を行ってきた。

この10年間に良く利用される50以上の原子力コードをベクトル化して蓄積してきた再構成のノウハウは、大きく分けると設計レベルとコーディングレベルの2種類のレベルがある。設計レベルでは、数値計算法やアルゴリズムから再考し並列処理向きのスキームやアルゴリズムの変更を行う。コーディングレベルは数値計算法自体の変更は行わずサブルーチン間の参照関係を考慮してDOループを移行したり、DOループの構造を変えたり、最適化のための制御文を挿入する等の技法（以下ベクトル化技法と呼ぶ）を適宜選択して行う。そのベクトル化の成果として、ベクトル化した50以上の原子力コードのベクトル化率は50%以下から平均で83%に向上し、また処理効率も平均で5倍改善されている⁷⁾。

さてこのベクトル化作業を振り返ってみると、ベクトル化の専門家とそれ以外の人では、ベクトル化作業の効率やベクトル化後の処理性能に大きな違いがある。例えば、軽水炉安全解析コードRELAP5は年を追うごとに機能追加やエラー修正がかかり、サブルーチン構造やDOループの構造などのプログラム構造が大きく変化している。ベクトル化の研究を始めたばかりでノウハウが少なかったころには、RELAP5/MOD1というバージョンをベクトル化するのに約20人月かかっていたのに対して、ノウハウが蓄積されてからベクトル化したRELAP5/MOD2ではベクトル化工数は10人月に半減した。⁷⁾この工数半減の原因は主に様々なコードで蓄積したベクトル化ノウハウによるベクトル化技法適用の試行錯誤的な繰り返し作業の減少であると考えられる。すなわち、原子力コードは、サブルーチン数、DOループやIF文の入れ子の数、ステートメント数が多く複雑な構造をしている場合が多々ある。このためベクトル化の専門家はプログラムの構造を良く理解し整理して適切なベクトル化技法の適用を検討する。ところが、ベクトル化経験の少ない者は不必要的ベクトル化技法の適用を検討する。ところが、ベクトル化経験の少ない者は不必要的ベクトル化技法の適用を検討したり、ベクトル化できなかった旨のメッセージに対する対処法がわからない場合があり、満足な処理効率の向上が得られないために試行錯誤の回数が多

くなりベクトル工数が専門家に比べて多い。

このようなベクトル化経験の有無によるベクトル化工数の差やベクトル化プログラムの処理性の差が小さくなるようにする方法として、ベクトル化専門家の持っている知識・ノウハウを整理してルール化したベクトル化支援エキスパート・システムが考えられる。ここでは、原研においてベクトル化の教育のために使用している例題⁸⁾のベクトル化を支援できる範囲に限定して、エキスパート・システム Ces-VP を試作して、ベクトル化経験の少ない者がベクトル化作業を効率良くできる方法としてエキスパート・システムを利用できるかどうかを検討した。

試作された Ces-VP システムは、ベクトル化診断、チューニング診断及び異常終了診断の機能を持っている。この Ces-VP では、プログラムの解析は利用者に任せているので、利用者は、ベクトル化コンパイラの出力リストやエラー・メッセージのリストを見ながらシステムの質問に答えながら診断をすすめていく。ベクトル化診断とチューニング診断では、利用者に対する質問によって DO ループ内で効率低下を起こしている部分を同定して効率向上につながるチューニングやベクトル化技法を推奨する。異常終了診断機能では、異常終了コード 240-0C6, 0C7について、エラー状況を質問し、プログラム中の調査すべき箇所を指摘する。最後に、Ces-VP を原研計算センター内で試用して、ベクトル化支援エキスパート・システムとしての有効性とマン・マシン・インターフェースについて調査した結果についても述べる。

本報告書の構成を以下に示す。第 2 章では Ces-VP の基になったベクトル化に関する知識について述べる。第 3 章では、Ces-VP のシステム概要について述べる。第 4 章では、実際に原研計算センターで行った Ces-VP を試用したアンケート調査の集計結果とその考察を行う。第 5 章では、本報告の結論を要約する。最後に付録として Ces-VP の核であるルールのリストを示す。

2. 原子力コードのベクトル化知識のルール化

Ces-VP を試作する目的は、原子力コードのベクトル化を支援するエキスパート・システムの実現性を検討することと、ベクトル化支援エキスパート・システムの問題点を明確にすることである。そのために、このシステムでは、約10年間に50本以上の原子力コードをベクトル化して蓄積してきた個々のコードに関する主なベクトル化知識を整理して盛り込むこととする。そこで今回の試作したベクトル化支援機能は、原研のベクトル化実作業担当者（主に計算センタ業務委託員）を対象に行ったベクトル計算機利用のための教育で使用した例題のベクトル化が支援できる範囲である。この例題は原子力コードのベクトル化に良く現れる技法と問題をまとめたものであり、試作システムの知識の範囲としては適当であると考えられる。また、このシステムの利用者は、FORTRAN 経験が十分でありベクトル化技法の意味は知っているがベクトル化経験は少ない人を想定している。

次にプログラムを前にしたときに専門家がどのように考えてベクトル化するのかということを解析して、教育資料に現れた例題からベクトル化のルールを抽出した方法について説明する。

Fig. 2.1 (a) で示すプログラムは、原子力コードでよく現れるプログラム構造を簡単化したものである。このプログラムの親ルーチン側では、繰り返し回数（以下ループ長と呼ぶ）がNの DO ループの中で主な計算を行うサブルーチン SUBX を呼び出している。この DO ループは、外部手続きが DO ループの中にあるためベクトル化できない。また、ループ長 N はベクトル化して効果があるような大きな数であり、計算ルーチン SUBX には DO ループがないためベクトル計算できないことがわかっているとする。そこで専門家はこのプログラムを見て次の 2 つのベクトル化技法を考える。

- ・ 親ルーチンの外部手続きの部分に計算ルーチンの内容を開発する。
- ・ 親ルーチン側の DO ループを計算ルーチン側に持ち込みベクトル化する。

このうちのどちらの方法を選択するかは、計算ルーチン SUBX の内容に依存する。すなわち、計算ルーチン SUBX のステートメント数が10行程度で変数名の整合が簡単にとれる場合には、前者の技法により親ルーチンの中に計算ルーチン SUBX を展開する。また原子力コードによくあるように、計算ルーチン SUBX のステートメント数が数千と多く、また使用変数名の整合をとることが困難であるとすると、後者の技法を使用して Fig. 2.1 (b) のようにプログラムをベクトル化する。この簡単な例題についても専門家は、次のようないくつかの点を判断材料にベクトル化の方法を決定していることがわかる。

- ・ DO ループは外部手続きが中にあるためベクトル化できない。
- ・ DO ループの繰り返し回数はベクトル化の効果のある大きさである。
- ・ 子ルーチンには、DO ループが無くベクトル化できない。
- ・ 子ルーチンのステートメント数

教育資料にある他の例についても、このように専門家が何に注目してベクトル化技法を選択しているかを考察した。

Fig. 2.2 (a) は配列 A に回帰参照の可能性があるためベクトル化できないプログラム例である。このプログラムを専門家が見たとき、まずベクトル化効果の出るループ長かどうか、さらにベクトル化表示およびベクトル化メッセージ JND-231 の意味を確認する。そして、コンパイラが配列 A が変数 M の値によって回帰参照になる可能性があると認識しベクトル化しないことに注目する。この時、常に $M \geq N$ または $M \leq 0$ で回帰参照がないことをわかっていれば、Fig. 2.2 (b) に示すように * VOCL LOOP, NOVREC 文を挿入することによって回帰参照のないことをコンパイラに指示する。これを整理すると、専門家は判断条件（以後選択パラメータと呼ぶ）として次の点に着目している。

- ・各 DO ループのループ長
- ・ベクトル化表示
- ・ベクトル化メッセージの番号
- ・回帰参照の有無

専門家は、これらの点について調べ、ベクトル化できる文を増加させるためにベクトル化技法として * VOCL LOOP, NOVREC 文⁹⁾ を挿入した。

Fig. 2.3 (a) は、内側ループ変数 J に関して回帰参照が起きているのでベクトル化できないプログラム例である。このプログラムを専門家が見たとき、ベクトル化効果の出るループ長か、また、ベクトル化表示とベクトル化メッセージ JND-262 によって内側 DO ループ変数 J に関して配列 A が回帰参照になっていることを確認する。この時、外側 DO ループ変数 I に関しては回帰参照でなく繰り返し回数 N が大きければ DO ループを入れ換えることを考え、DO ループの入れ換えによってデータの参照関係に矛盾がないことを確認した後再構成する。また配列 A, B へのアクセスは等間隔から連続アクセスとなり効率が良くなる。これを整理すると、専門家は選択パラメータとして次の点に着目している。

- ・各 DO ループのループ長
- ・ベクトル化表示
- ・ベクトル化メッセージの番号
- ・DO ループの多重度
- ・配列の添字変数
- ・参照関係の矛盾の有無
- ・回帰参照の有無

専門家は、これらの点について調べ、ベクトル化できる文の増加およびメモリアクセスの向上という観点でベクトル化技法として DO ループの入れ換えという技法を選択した。

Fig. 2.4 (a) は入出力文があるためベクトル化できないプログラム例である。このプログラムを専門家が見た時ベクトル化効果の出るループ長かどうか、またベクトル化表示やベクトル化メッセージによってベクトル化非対象文である WRITE 文の存在および回帰参照の可能性のある配列があることを確認する。そして WRITE 文がデバッグ等で使用される不要なものでなければ回帰参照ないので WRITE 文の前後で DO ループの分割をする。これを整理すると専門家は選択パラメータとして次の点に着目している。

- ・各 DO ループのループ長

- ・ベクトル化表示
- ・ベクトル化メッセージの番号
- ・回帰参照の有無
- ・コメント化の可能性

専門家はこれらの点について調べ、ベクトル化可能な文と不可能な文の分離及びベクトル化できる文の増加という観点でベクトル化技法としてDO ループの分割という技法を選択した。

Fig. 2.5 (a) は、すでにベクトル化されている一重 DO ループのプログラム例である。このプログラムを見た時、ベクトル化表示によってすでに全ての文がベクトル化されていることを確認してさらにベクトル処理効率を向上させることを考えるが一重 DO ループであり、ループ長がベクトル効果の出る大きさでしかもその最大値が既知であるとすると、Fig. 2.5(b) に示すように *VOCL LOOP, REPEAT 文⁹⁾の挿入によってベクトルレジスタの構成を最適に設定することを考える。これを整理すると専門家は選択パラメータとして次の点に着目している。

- ・ベクトル長の大きさ
- ・ベクトル長の最大値

専門家は、これらの点について調べ、ベクトル・レジスタの有効利用という観点でベクトル化技法として最適化制御文を挿入する技法を選択した。

このようにして選択パラメータが抽出され選択パラメータの状態によって様々な推奨技法が選択されるようにルールを作成した。Ces-VP の推奨する技法とその選択パラメータを Table 2.1 に示す。この表は、1つの技法が推奨されるときの選択パラメータを列挙したものである。選択パラメータは全部で22種類あり、その内容は表の下部に示してある。

次に Ces-VP のもう1つの機能である異常終了診断に関するルールについて述べる。これは、スカラー処理では問題にならないが VP システムを利用してプログラムを実行した時にのみ発生する異常終了（例えば、非正規化データと8バイト境界にない倍精度データの扱い）に対する原因と対処を推奨するものである。支援する範囲は、ベクトル化やチューニング診断と同様に教育資料の例題（異常終了コード 240-0C6 および 0C7）についての診断が可能である。次に、専門家がどのような点に注目して異常終了の原因を特定し対処するかを解析して、ルールがどのように抽出されたかについて述べる。

異常終了コード 240-0C6 は命令例外のために起こり、その原因としては次のような事が考えられる。

- ・8バイト境界にない倍精度データをベクトル命令で使用
- ・ベクトル命令の使用法の誤り

Fig. 2.6 は倍精度データが8バイト境界にないために異常終了コード 240-0C6 で異常終了しているプログラム例と異常終了時に出力されたトレースバック情報を示している。専門家は異常終了した時そのトレースバック情報から異常終了コードとエラーが起きた場所を確認する。この例では、トレースバック情報からエラーはサブルーチン SUB1 の ISN 5 番の CONTINUE 文で起きたと考える。ところが専門家は、コード 0C6 は倍精度データが8バイト境界にないことが主な原因であると知っているので、実際には CONTINUE 文ではなくその前の文で現れる倍精度データつまり配列 A のアドレス設定に関連する部分に注目する。配列 A は、仮引数になっているの

で上位ルーチンの MAIN プログラムの CALL 文に注目する。ここでトレースバック情報を見直し MAIN プログラムの ISN 5 番の CALL 文の実引数 IB に注目する。整数型 4 バイト配列 IB は実数型 8 バイトの変数 X と EQUIVALENCE 文によって結合され、また、変数 X は 8 バイト境界上に作成されるために、IB の偶数添字の配列要素は 8 バイト境界上にできる。IB(1) を実引数として使うと実数型 8 バイトである仮引数 A は 8 バイト境界上にない倍精度データとなり異常終了が起こる。これを整理すると異常終了コード 240-0C6 の場合、次のようなことが判断条件となって異常終了の原因を特定していることがわかる。

- ・エラーデータが仮引数に存在
- ・エラーデータは実数型 8 バイトでその実引数は整数型 4 バイト
- ・実引数は EQUIVALENCE 文によって他の実数型 8 バイトの変数と結合

次に、ベクトル命令の使用法の誤りによって終了コード 240-0C6 で異常終了する例を Fig. 2.7 に示す。専門家はトレースバック情報を見て SUBX の DO200 の部分に注目する。この時、まず第 1 に DO ループの前に *VOCL LOOP, REPEAT 文があるのでその繰り返し回数を確認する。この場合、実際の繰り返し回数は N の実引数から 600 であることがわかる。しかし、*VOCL LOOP, REPEAT 文では 10 回しか繰り返されないとしてコンパイラに指示しているのでこの指定回数が異常終了の原因と特定する。この場合の判断条件は、次の 2 つである。

- ・*VOCL LOOP, REPEAT 文が存在
- ・*VOCL LOOP, REPEAT 文で指示した繰り返し回数が実際の繰り返し回数より小さい

このように専門家はコード 240-0C6 で異常終了した場合に、どのような命令例外が起こったのかを調べる。そこで、第 1 に *VOCL LOOP, REPEAT 文の存在を確認して、あれば指示した繰り返し回数が実際の繰り返し回数を越えていないかを確認する。越えていれば *VOCL LOOP, REPEAT 文が原因だと考える。*VOCL LOOP, REPEAT 文がないかまたは *VOCL LOOP, REPEAT 文の指示した繰り返し回数が実際の繰り返し回数以下であれば、異常終了をおこしたデータが仮引数であるか、または、EQUIVALENCE 文で定義されているかを確認し、8 バイト境界にデータがあるかどうかを調べる。以上のような事を、判断条件という観点でまとめてみると Table 2.2 のようになる。Table 2.2 は異常終了コード 240-0C6 で推奨する対策毎に原因を特定するための判断条件を示している。図中の○は判断条件の肯定、×は否定を表している。条件は a～e で表現され、a から e の方向で条件を吟味していく原因を特定する。b の欄の - は *VOCL LOOP, REPEAT 文の存在を仮定しているため a が × のときには意味のないことを示している。また *VOCL LOOP, REPEAT 文の修正では b の段階で原因が特定されるので c 以降は空欄になっている。

次に、非正規化データを使用することにより起こる終了コード 240-0C7 で異常終了しているプログラム例を Fig. 2.8 に示す。コード 0C6 のときと同様に専門家はトレースバック情報から異常終了コードの C7 と異常終了がおきた場所を SUB2 の ISN 5 番と確認する。この場合は、CONTINUE 文であるが C7 が非正規化データの存在を表していることを知っているので ISN 5 番以前でベクトル計算している実数型のデータを探し、ISN 4 番の配列 RA に注目する。RA は SUB2 で更新されない整合配列なので上位ルーチンの MAIN の ISN 8 番に視点を移す。そして RA とその実引数である IA の宣言型について調べ整数型 4 バイトの配列であることを知る。

RAは実数型8バイトなので引数の型が一致していないことがわかる。そしてMAINのISN6番でRAに渡されるIAの要素に0でない値が代入されていることに注目する。これはSUB2でRAが実数型であるのに整数型のデータ形式で値が設定されていることを表す。VPシステムでは、ベクトル演算で実数型の変数の和を求める際には正規化されたデータ形式を持っていなければならない。したがってRAが整数型のデータ形式をもっていることが異常終了の原因となっていると判断する。これを整理すると異常終了コード240-0C7の場合、次のようなことが判断条件となっていることがわかる。

- ・サブルーチンSUB2では異常終了したデータRAが更新されていない。
- ・サブルーチンSUB2の仮引数にRAがある。
- ・異常終了した実数型8バイトデータに対応する実引数は整数型4バイトである。
- ・IAには0以外の値が代入されている。

これらの条件によって原因は整数型の変数IAと実数型の変数RAが結合され、かつ、IAに0以外の値が代入されたことであることが結論され、引数の型等の再検討を推奨する。

Table 2.1 Vectorization technique recommended by Ces-VP and selection parameters

推奨技術	選択パラメータ
① 下位ルーチンへのDOループの持ち込み	a, b, e, n, v
② 最内DOループの書き下し	a, b, v
③ DOループの分割	a, b, e, f, g, o, s, t
④ DOループの入れ換え	a, b, e, i, s, t, v
⑤ *VOCL LOOP, NOVREC文の挿入	a, b, c, t
⑥ アンローリング	a, b, v
⑦ 多次元配列の第1添字の変更	a, b, g, h, i
⑧ DO変数の増分値の変更	a, c
⑨ *VOCL LOOP, REPEAT文の挿入	d
⑩ *VOCL STMT, IF(真率)の挿入	l
⑪ IF文の入れ子の統合	k
⑫ リストベクトルの作成	g, i, k
⑬ 中間配列のスカラー変数化	a, u
⑭ 上位ルーチンでのインライン展開	a, b, e, n, p, q, v
⑮ DOループの統合	a, r, j, s
⑯ DOループの1重化	a, b, i, s, v
⑰ ベクトル処理とスカラー処理の場合分け	a, b
⑲ *VOCL LOOP, SCALAR文の挿入	a, b, j
⑳ デバッグ文のコメント化	a, b, e, o
㉑ 入出力文中の計算式の独立	a, b, e, o
㉒ 文字型変数及び配列の実数型への型変換	a, b, e
㉓ メッセージに出力された変数の作業配列化	a, b, c

< 選択パラメータの内容 >

a 各DOループのループ長	h 配列の宣言タイプ・数	o コメント化の可能性
b ベクトル化表示	i 配列の添字変数	p ある文を境にした分割可能性
c DO変数の増分値	j IF文の数	q 上位ルーチンでの呼び出し状況
d ベクトル長の最大値	k IF文の入れ子の数	r隣接したDOループの構造
e ベクトル化メッセージの番号	l IF文の真率	s 参照関係の矛盾性
f プログラムのステートメント数	m リストベクトルの使用	t 回帰参照の有無
g GOTO文の飛び先・数	n 下位ルーチンの構造	u 作業配列の用途
		v DOループの多重度

Table 2.2 Reasoning table for abend code OC6

判 断 条 件		R E P E A T 文 の 有 無	R E P E A T 文 の 繰 り 返 し 回 数	異 常 終 了 上 位 デ ル タ は 仮 引 で 数 定 に 義 あ る	仮 実 引 数 数 は は 実 4 バ 型 イ 8 ト バ イ 数 ト 型 で ・ 整 か 数 つ 型	実 E 引 Q 数 U が I 他 V の A 実 L 数 E 型 N 8 C バ E イ 文 ト で 変 結 数 合 と	
異常終了に対する 推奨対策							
* VOC L LOOP, REPEAT文の繰り返し回数の修正	○	○					
実引数と結合しているEQUIVALENCE 文の修正	×	—	○	○	○		
実引数と実数型8バイトの変数をEQUIVALENCE 文で結合する	○	×	○	○	○	×	

(a) original program

```

S      DO 10 I=1,N          親ルーチン
S          CALL SUBX( A(I), B(I), S1, S2 )
S 10  CONTINUE

SUBROUTINE SUBX( A, B, S1, S2 )      計算ルーチン
A = S1*B + S2

数千ステートメント

RETURN
END

```

(b) vectorized program

```

CALL SUBX( A, B, S1, S1, N )          親ルーチン

SUBROUTINE SUBX( A, B, S1, S2, N )      計算ルーチン
DIMENSION A(N),B(N)
DO 10 I=1,N
A(I) = S1*B(I) + S2

数千ステートメント

V 10  CONTINUE
RETURN
END

```

Fig. 2.1 Vectorization by transfer of DO loop between procedures.

(a) original program

```

C
C *** EXAMPLE OF *VOCL LOOP, NOVREC
C
SUBROUTINE NOVREQ( A , B , N , M )
IMPLICIT REAL*8( A-H , O-Z )
DIMENSION A(N+M),      B(N)
C
S   DO 10 J=1,N
S       A(J+M) = A(J)*B(J)
S   10 CONTINUE
C
      RETURN
END

```

(b) vectorized program

```

C
C *** EXAMPLE OF *VOCL LOOP, NOVRBC
C
SUBROUTINE NOVREQ( A , B , N , M )
IMPLICIT REAL*8( A-H , O-Z )
DIMENSION A(N+M),      B(N)
C
*VOCL LOOP, NOVREC
V   DO 10 J=1,N
V       A(J+M) = A(J)*B(J)
V   10 CONTINUE
C
      RETURN
END

```

Fig. 2.2 Vectorization by specifying the relationship of the size subscript of array element.

(a) original program

```

C
C *** EXAMPLE OF LOOP INDEX EXCHANGE 1 ***
C
SUBROUTINE LOOPX( A , B , N , M )
IMPLICIT REAL*8( A-H , O-Z )
DIMENSION A(N,M),      B(N,M)
C
S   DO 10 I=1,N
S   DO 10 J=1,M-1
S       A(I,J+1) = A(I,J)*B(I,J)
S 10  CONTINUE
C
RETURN
END

```

(b) vectorized program

```

C
C *** EXAMPLE OF LOOP INDEX EXCHANGE 1 ***
C
SUBROUTINE LOOPX( A , B , N , M )
IMPLICIT REAL*8( A-H , O-Z )
DIMENSION A(N,M),      B(N,M)
C
S   DO 10 J=1,M-1
V   DO 10 I=1,N
V       A(I,J+1) = A(I,J)*B(I,J)
V 10  CONTINUE
C
RETURN
END

```

Fig. 2.3 Vectorization by changing the order of DO loops for removal of recursive reference.

(a) original program

```

C
C *** EXAMPLE OF LOOP DISTRIBUTION 2 ***
C
SUBROUTINE WRITE( A , B , C , S1 , S2 , S3 , N )
IMPLICIT REAL*8( A-H , O-Z )
DIMENSION A(N), B(N), C(N)
C
DO 10 I=1,N
      A(I) = S1*C(I)
      B(I) = S2*C(I)
      WRITE( 6 , '(1H , 2D15.7)' ) A(I), B(I)
      A(I) = A(I) + S3*B(I)
10   CONTINUE
C
      RETURN
END

```

(b) vectorized program

```

C
C *** EXAMPLE OF LOOP DISTRIBUTION 2 ***
C
SUBROUTINE WRITE( A , B , C , S1 , S2 , S3 , N )
IMPLICIT REAL*8( A-H , O-Z )
DIMENSION A(N), B(N), C(N)
C
DO 10 I=1,N
      A(I) = S1*C(I)
      B(I) = S2*C(I)
10   CONTINUE
      WRITE( 6 , '(1H , 2D15.7)' ) (A(I),B(I),I=1,N)
DO 15 I=1,N
      A(I) = A(I) + S3*B(I)
15   CONTINUE
C
      RETURN
END

```

Fig. 2.4 Vectorization by separating the DO loop at the I/O statements.

(a) original program

```

C
C     *** EXAMPLE OF *VOCL LOOP, REPEAT 2 ***
C
C     SUBROUTINE VOCLRE( A , B , C , D , E , F , G , H ,
&                      O , P , Q , R , S , T , U , V , W , N )
C     IMPLICIT REAL*8( A-H , O-Z )
C     DIMENSION A(N),      B(N),      C(N),      D(N),      E(N),
D          G(N),      H(N),      O(N),      P(N),      Q(N),
D          S(N),      T(N),      U(N),      V(N),      W(N)
C
V     DO 10 I=1,N
V         A(I) = 0.25D0*(B(I)+C(I)+D(I)+E(I))
&           + 0.25D0*(F(I)+G(I)+H(I)+O(I))
&           + 0.25D0*(P(I)+Q(I)+R(I)+S(I))
&           + 0.25D0*(T(I)+U(I)+V(I)+W(I))
V   10  CONTINUE
C
C     RETURN
C     END

```

(b) vectorized program

```

C
C     *** EXAMPLE OF *VOCL LOOP, REPEAT 2 ***
C
C     SUBROUTINE VOCLRE( A , B , C , D , E , F , G , H ,
&                      O , P , Q , R , S , T , U , V , W , N )
C     IMPLICIT REAL*8( A-H , O-Z )
C     DIMENSION A(N),      B(N),      C(N),      D(N),      E(N),
D          G(N),      H(N),      O(N),      P(N),      Q(N),
D          S(N),      T(N),      U(N),      V(N),      W(N)
C
*VOCL LOOP, REPEAT(128)
V     DO 10 I=1,N
V         A(I) = 0.25D0*(B(I)+C(I)+D(I)+E(I))
&           + 0.25D0*(F(I)+G(I)+H(I)+O(I))
&           + 0.25D0*(P(I)+Q(I)+R(I)+S(I))
&           + 0.25D0*(T(I)+U(I)+V(I)+W(I))
V   10  CONTINUE
C
C     RETURN
C     END

```

Fig. 2.5 Vector optimization by specifying vector length.

```
000001      INTEGER IA(100),   IB(101)
000002      REAL*8 X
000003      EQUIVALENCE ( X, IA(1), IB(2) )
000004      C
000004      CALL SUB1( IA , 50 )
000005      C
000005      CALL SUB1( IB , 50 )
000006      C
000006      STOP
000007      END
```

```
000001      SUBROUTINE SUB1( A , N )
000002      REAL*8 A(*)
000003      V      DO 10 I = 1, N
000004      V          A(I) = 1.0D0
000005      V      10    CONTINUE
000006      RETURN
000007      END
```

JZL240I-U (ABEND) CODE=S00C6-U0000 PSW=078D0000A0101742 SDWA=0011C71C
ADVANCED MODE VCR=3D000032
UNPERMITTED INSTRUCTION INSTRUCTION=892000031AA2 LOCATION=0010173E
GR0 =FD000016 GR1 =0010132C GR2 =00000190 GR3 =00000031
GR4 =00000001 GR5 =0010132C GR6 =00000032 GR7 =00101314
GR8 =00000200 GR9 =00000032 GR10=00000008 GR11=00101314
GR12=00000001 GR13=00101588 GR14=60101510 GR15=00101520
FR0 =4110000000000000 FR2 =0000000000000000
FR4 =4110000000000000 FR6 =4B00000000FB944C
FR8 =006800AC007C00C4 FR10=009000DC00A400F4
FR12=00B8010C00CC0124 FR14=00E0013C00F40154
ERROR OCCURS AT SUB1 ISN 00005 LOC 0010173E OFFSET 00021E
SUB1 AT LOC 00101520 CALLED FROM LOC 6010150E IN MAIN AT ISN 00005
MAIN AT LOC 401011EA CALLED FROM (0, \$)

Fig. 2.6 An example of abend code OC6 by double precision data not no 8-byte boundary.

```

000001      REAL*8      A(600), B(600)
000002  V      DO 100 I = 1 , 600
000003  V      A(I) = DBLE( I-1 )/600.00
000004  V  100 CONTINUE
000005      CALL SUBX( A , B , 600 )
000006      STOP
000007      END

```

```

000001      SUBROUTINE SUBX( A , B , N )
000002      REAL*8  A(*), B(*)
000003      *VOCAL LOOP, REPEAT(10)
000004  V      DO 200 I = 1 , N
000005  V      B(I) = DSIN( 3.1415*A(I) )
000006  V  200 CONTINUE
000007      RETURN
000007      END

```

JZL2401-U (ABEND) CODE=S00C6-U0000 PSW=078D0000E01040C4 SDWA=0011F71C
ADVANCED MODE VCR=3D000200
UNPERMITTED INSTRUCTION INSTRUCTION=FFA00008B010 LOCATION=001040BE
GR0 =FD000016 GR1 =00101730 GR2 =00000256 GR3 =00000008
GR4 =00000000 GR5 =00000018 GR6 =00102A10 GR7 =00101750
GR8 =00000008 GR9 =00000258 GR10=0010172C GR11=00101738
GR12=001029F8 GR13=00103E70 GR14=60103DF4 GR15=00103E08
FR0 =3E603A06D3A06D3A FR2 =00AFC93C00FDE08C
FR4 =00AFCCE8480AFCFB0 FR6 =0064010000000000
FR8 =000000003F120000 FR10=00AFC838000000008
FR12=00AE461000000000 FR14=00AFC94800AE4610
ERROR OCCURS AT SUBX ISN 00005 LOC 001040BE OFFSET 0002B6
SUBX AT LOC 00103E08 CALLED FROM LOC 60103DF2 IN MAIN AT ISN 00005
MAIN AT LOC 401015EA CALLED FROM (O.S)

Fig. 2.7 An example of abend code OC6 by specifying the incorrect number of REPEAT.

```

000001      INTEGER*4  IA(100)
000002      REAL*8     A(50), RA
000003      EQUIVALENCE ( IA(1), RA)
C
000004  V      DO 10 I = 1, 99, 2
000005  V          IA( I ) = 0
000006  V          IA(I+1) = 123456789
000007  V      10 CONTINUE
C
000008      CALL SUB2( IA , A , 50 )
C
000009      STOP
000010      END

000001      SUBROUTINE SUB2( RA , A , N )
000002      REAL*8 RA(*), A(*)
000003  V      DO 10 I = 1, N
000004  V          A(I) = RA(I)+RA(I)
000005  V      10 CONTINUE
000006      RETURN
000007      END

```

JZL240I-U (ABEND) CODE=S00C7-U0000 PSW=078D0000E010274C SDWA=0011C71C
ADVANCED MODE VCR=30000200
UNPERMITTED INSTRUCTION INSTRUCTION=FEB000000000 LOCATION=00102746
GR0 =FD000016 GR1 =00102124 GR2 =00000190 GR3 =00000031
GR4 =001022C8 GR5 =00000032 GR6 =00102118 GR7 =001022B0
GR8 =00000200 GR9 =00000200 GR10=00000198 GR11=00102120
GR12=00000000 GR13=00102560 GR14=401024EC GR15=001024F8
FR0 =16E9ACE400AFC0F0 FR2 =0000000060000000
FR4 =00000000C4404040 FR6 =E2F3F5C440404040
FR8 =00AB56B000AB5A98 FR10=00CBA83000E9A6E0
FR12=C003000000CBAB50 FR14=00AFC144E6E3D600
ERROR OCCURS AT SUB2 ISN 00005 LOC 00102746 OFFSET 00024E
SUB2 AT LOC 001024F8 CALLED FROM LOC 401024EA IN MAIN AT ISN 00008
MAIN AT LOC 40101FC2 CALLED FROM (0, S)

Fig. 2.8 An example of abend code OC7 by using the nonnormalized data.

3. Ces-VPシステムの概要

第2章で見たように専門家がプログラムを見たときにどのように考えてベクトル化しているかを解析して、ルールを作成した。本章では、これらのルールをどのようにしてエキスパート・システムとして実現するかという方法について記述する。

3.1 機能概要

Ces-VPの大きな機能としては次の2点がある。

- ベクトル化・チューニング診断機能

今回の試作では、利用者に対する質問によってDOループ内で効率低下を起こしている部分を同定し効率向上に繋がるチューニング・ベクトル化技法を推奨する。

- 異常終了処理診断機能

今回の試作では、異常終了コード240-OC6, OC7についてエラー状況に対する質問をして、プログラム中の調査すべき箇所を指摘する。

3.2 エキスパート・システム構築ツール

Ces-VPを作成するのには、富士通のエキスパート・システム構築ツールESHELL¹⁰⁾を使用している。このESHELLはUTILISP下で動作し、システムとして知識を活用して解決を行う推論エンジン（推論機構）と特定分野の知識や経験則を蓄える知識ベースを編集する知識エディタ（構築支援機構）から構成される。Fig. 3.1にESHELLを使ってエキスパート・システムを構築する場合のエキスパート・システム本体とその作成者や利用者の位置づけを示す。作成者は知識エディタとの会話処理によってユーザ・プログラムの部分（推論エンジンの制御機構・知識源・黒板・フレーム型知識ベース・ユーザ定義関数）を構築する。また、利用者はユーザ定義関数を介して構築したシステムから質問を受けたり、それに対する回答を入力したり推論結果を見ることができる。

Fig. 3.2にESHELLのシステム構成を示す。推論エンジンは後述するような前向き推論・後戻り推論および推論過程の説明を行うツールを持っている。知識エディタは、デバッガ・デザイナ・ブラウザ・テッカ・フレームの5つのユティリティを用意していて利用者はガイドメニューに従って、会話的に作業を進めていくことができるようになっている。エキスパート・システムを作成する者は、知識エディタを利用して推論エンジンの制御方法や問題領域の知識等を記述すればよい。ESHELLでは、システム作成者が記述するこれらの部分をユーザ・プログラムと呼ぶ。またユーザプログラムは、2種類の形式の知識表現、即ち、if., then., elseの型で書かれたプロダクション型知識ベースと構造を持った静的な知識の集まりであるフレーム型知識ベース、が可能である。プロダクション型知識ベースには、対象とする問題を解決するための知識

(知識源) の他に推論のためのデータ格納領域（黒板）や推論の制御方式（制御機構）に関する記述が必要となる。また利用者固有の手続きや処理はユーザ定義関数で定義する。

ESHELL では、推論エンジンによって知識ベースの実行が制御されている。推論エンジンはまず初期処理を行い最初の知識源（ルールの集まり）を起動する。そして起動された知識源で適合するルールを実行する。ルールの実行のしかたによって前向き推論や後戻り推論等の推論方式を変えることができる。この時、ルールの右辺が実行され何かのイベントを発行するとイベントキューという一種のスタックに登録される。このイベントには、起動したい知識源の情報が書かれている。次に推論を終了するかどうかの判定をして、終了しないならば次の知識源を選択し起動する。この選択は、イベントキューから取り出されるが、この取り出し方によって探索方法を変えることができる。このような知識源の選択・起動・知識源内のルールの実行・終了判定という4つの処理を、終了判定で終了と判定されるまで繰り返される。

上で述べたイベントを発行する場合には、イベント名と呼ばれる起動される知識源のトリガ名と起動される知識源の使用するノード名を常にペアで指定する。このペアはイベントエントリと呼ばれる。ここでトリガ名とは、知識源に複数指定できる名前であり、同じトリガを持つ知識源は同時に起動されることになる。ノードとは、ルールの実行で参照や更新がされる黒板上のデータの部分集合で、その名前がノード名である。つまりイベントエントリとは評価するルールの集合とその時使用するデータの集合を定義しているわけである。そしてイベントキューにはこのイベントエントリが登録されている。イベントキューは、スタックであり、常に発行された順に先頭から登録される。取り出すときには、先頭から取り出すと深さ優先探索、最後尾から取り出すと広さ優先探索が実現される。Fig. 3.3 は、深さ優先探索の場合と広さ優先探索の場合の探索順序を示す図である。エントリ 11 の KS では、エントリ 23, エントリ 22, エントリ 21 の順番でイベントを発行し、エントリ 21 では、エントリ 32, エントリ 31 の順番でイベントを発行する。またエントリ 22 はエントリ 33 のイベントを発行する。矢印は探索する順序である。次にイベントキューの登録と取り出しの状態を説明する。先ず、Fig. 3.3 (a) で、深さ優先探索の説明をする。

<イベントキューの状態>

初期状態 : [空]

最初の KS が実行される。

エントリ登録 : [エントリ 11]

エントリ取り出し : [空]

エントリ 11 の KS が実行され、エントリ 23, エントリ 22, エントリ 21 が発行される。

エントリ登録 : [エントリ 21, エントリ 22, エントリ 23]

エントリ取り出し : [エントリ 22, エントリ 23]

①先頭のエントリ 21 の KS が実行され、エントリ 32, エントリ 31 が発行される。

エントリ登録 : [エントリ 31, エントリ 32, エントリ 22, エントリ 23]

エントリ取り出し : [エントリ 32, エントリ 22, エントリ 23]

②エントリ 31 の KS が実行される。

エントリ登録 : [エントリ 32, エントリ 22, エントリ 23]

エントリ取り出し : [エントリ 22, エントリ 23]

③エントリ32のKSが実行される。

エントリ登録 : [エントリ22, エントリ23]

エントリ取り出し : [エントリ23]

④エントリ22のKSが実行され、エントリ33が発行される。

エントリ登録 : [エントリ33, エントリ23]

エントリ取り出し : [エントリ23]

⑤エントリ33のKSが実行される。

エントリ登録 : [エントリ23]

エントリ取り出し : [空]

⑥エントリ23のKSが実行される。

エントリが空でエントリ23のKSでもイベントが発行されないので終了する。

次に、Fig. 3.3 (b) で、広さ優先探索の説明をする。

<イベントキューの状態>

初期状態 : [空]

最初のKSが実行される。

エントリ登録 : [エントリ11]

エントリ取り出し : [空]

エントリ11のKSが実行され、エントリ23, エントリ22, エントリ21が発行される。

エントリ登録 : [エントリ21, エントリ22, エントリ23]

エントリ取り出し : [エントリ21, エントリ22]

①最後のエントリ23のKSが実行される。

エントリ登録 : [エントリ21, エントリ22]

エントリ取り出し : [エントリ21]

②エントリ22のKSが実行される。

エントリ登録 : [エントリ33, エントリ21]

エントリ取り出し : [エントリ33]

③エントリ21のKSが実行され、エントリ32, エントリ31が発行される。

エントリ登録 : [エントリ31, エントリ32, エントリ33]

エントリ取り出し : [エントリ31, エントリ32]

④エントリ33のKSが実行される。

エントリ登録 : [エントリ31, エントリ32]

エントリ取り出し : [エントリ31]

⑤エントリ32のKSが実行される。

エントリ登録 : [エントリ31]

エントリ取り出し : [空]

⑥エントリ31のKSが実行される。

エントリが空でエントリ31のKSでイベントが発行されないので終了する。

次に前向き推論と後戻り推論という推論方式の説明を行う。前向き推論(Forward 推論)とは与えられた事実から出発して適用可能なルールを探し事実を増やしていくことで推論をすすめ結論を導く方法である。例として次のような事実と規則が与えられているものとする。

- | | |
|------|-----------------|
| 規則群： | 規則① 「BならばCである。」 |
| | 規則② 「CならばDである。」 |
| | 規則③ 「DならばEである。」 |
| 事実群： | 事実① 「AならばBである。」 |
| | 事実② 「FならばGである。」 |

この時、「AならばEであるか?」の質問に対して前向き推論では次のように推論を進める。

- I. 事実①と規則①から「AならばCである」と推論する。
- II. Iの中間結果(新しい事実)と規則②から「AならばDである。」と推論する。
- III. さらにIIの中間結果と規則③から「AならばEである。」と結論する。

後戻り推論(Backchain)では、ある仮説を推論目標としてその仮説が成立するための前提条件を再帰的に調べていき、その仮説が立証できるかを結論する方法である。前向き推論と同じ例で推論過程を説明する。

- I. 質問「AならばEであるか?」を「AならばEである。」と仮定する。そしてEが結論部にある規則③を探し出して「AならばDである。」が成立すれば仮説は成立すると推論する。
- II. 既知の事実には「AならばDである。」がないので、さらにこれを仮説としてIと同様にDが結論部にある規則を探し「AならばCである。」が成立すれば「AならばEである。」が成立すると推論する。
- III. 同様に「AならばBである。」が成立すれば推論目標の仮説が成立することを推論する。
- IV. IIIの仮説は事実①より成立するので推論目標の「AならばEである。」の仮説が立証される。

3.3 Ces-VP の構成

Ces-VP は Fig. 3.4 のように、プロダクション・ルール 206 個を集めた 24 個の知識源、8 個のレベルを持つ黒板、推論エンジン、主にマン・マシン・インターフェースを向上するために新しく定義した 58 個の LISP 関数から構成されている。以下にその概要を示す。各知識源のルールの詳細は付録で示す。

(1) 知識源 (KS)

知識源の制御ツリーを Fig. 3.5 に示す。ベクトル化・チューニング診断部分は、診断内容選択 KS、ベクトル化・チューニング制御 KS、チューニング診断 KS、ベクトル化診断制御 KS、ベクトル化メッセージ診断制御 KS、ベクトル化メッセージ別の診断 KS、ベクトル化・チューニング結果出力 KS の 7 種類から構成されている。異常終了処理診断部分は、異常終了処理制御 KS、異常終了コード 0C6 診断 KS、0C6 結果出力 KS、異常終了コード 0C7 診断 KS、C7 探索制御 KS、C7 元データ探索 KS、C7 EQUIVALENCE データ検索 KS、C7 バックトラ

ック KS, C7 診断制御 KS, C7 結果出力 KS の10種類から構成されている。以下では各 KS の概要を述べる。

a. 診断内容選択 KS (ルール数 3 個)

初期メニューの制御で 1 つの DO ループに関してのコンサルトが終了または必ず初期メニューに戻るようにしてある。

b. ベクトル化・チューニング制御 KS (ルール数 8 個)

使用方法の説明、黒板の初期設定、ベクトル化診断とチューニング診断の制御を行う。ベクトル・コンパイラの出力するソース・プログラムで対象 DO ループ内の全ての実行文がベクトル化できない場合はベクトル化診断 KS, 全てがベクトル化されている場合はチューニング診断 KS, ベクトル化している部分とされていない部分が混在する場合は両方の KS が発行される。

c. チューニング診断 KS (ルール数 23 個)

チューニング診断をして推奨技法のフラグを設定する。

d. ベクトル化診断制御 KS (ルール数 3 個)

ベクトル化メッセージが出力されていれば、ベクトル化診断のためのベクトル化メッセージ診断制御 KS を発行し、また、チューニング診断 KS が DO ループの多重度、ベクトル長の大きさ、IF 文の有無によって発行される。

e. ベクトル化メッセージ診断制御 KS (ルール数 13 個)

各ベクトル化メッセージに対する KS を発行する。

ベクトル化している旨のメッセージの場合はチューニング診断 KS を発行する。

f. 各ベクトル化メッセージ診断 KS (ルール総数 29 個)

メッセージ番号 JND-227, 228, 229, 231, 249, 250, 262, 273, 277 に対してそれぞれ 1 つずつの KS があり、推奨技法のフラグを設定する。

g. ベクトル化・チューニング結果出力 KS (ルール数 62 個)

ベクトル化及びチューニング診断で設定されたフラグに基づいて診断結果を画面に表示する。

h. 異常終了処理制御 KS (ルール数 5 個)

スカラー処理で正常に終了しているかを確認して異常終了コードによって OC6 か OC7 の診断 KS を発行する。

i. C6 診断 KS (ルール数 6 個)

C6 の対策フラグを設定する。

j. C6 結果出力 KS (ルール数 6 個)

C6 の対策を表示する。

k. C7 診断制御 KS (ルール数 4 個)

異常終了したデータに対する情報を利用者に質問して設定する。

l. C7 診断制御 KS (ルール数 4 個)

異常終了したデータが EQUIVALENCE 文で他の変数と結合されている場合の処理および非正規化データの検索 KS を発行する。

m. C7 元データ探索 KS (ルール数 15 個)

利用者の質問に対する答から、非正規化データのエラー原因設定を行う。また、上位ルーチン

でデータが定義されると利用者が答えた場合は C7 診断制御 KS を発行しそのデータの情報を定義する。また、そのデータが間接的に他の変数で定義されているまたは COMMON 変数で定義されると利用者が答えた場合もその定義がどのような値でされているかを設定し判断する。異常終了したルーチンより上位にあるルーチンで対象データが非正規化データの原因にならないと判断するとバックトラックする KS を発行する。この場合のバックトラックとは、非正規化データの原因になっていると思われるすべてのデータについてその定義値と宣言型が調べられるが、この時非正規化の原因と思われるようなデータでない場合に元の下位ルーチンに戻り、他の候補のデータについて調べていくことを意味する。

n. C7 EQUIVALENCE データ検索 KS (ルール数 15個)

C7 元データ探索と同様に EQUIVALENCE 文で結合されたデータに対して検索する。

o. C7 バック・トラック KS (ルール数 8 個)

引数で結合されている場合のデータが非正規化データの原因でない場合のバック・トラックを行う。

p. C7 結果出力 KS (ルール数 2 個)

C7 のエラーの原因とそのサブルーチン名・データ名を表示する。

(2) 黒板 (BB)

黒板は、推論するための入力データや推論の中間結果などを保持するためのデータ領域で、レベル、ノード、アトリビュートの順に階層構造になっている。レベルは、ノードの定義体であり複数のノードを定義出来る。ノードはデータの格納領域であり、その下に実際のデータであるアトリビュートと他のノードとのポインタであるリンクを複数定義することができる。ノードにはスタティック・ノードとダイナミック・ノードがある。これらは、生成のされ方が違いスタティック・ノードが推論の開始に先立ち、レベルで定義されている通りに生成されるのに対してダイナミックノードは推論の途中でルールの右辺で指示して動的に生成される。これは、レベルを雛形にしていくつも生成することができる。ベクトル化・チューニング診断部分では、DO ループの基本情報レベル、ベクトル化に関する質問レベル、ベクトル化診断レベル、推奨技法フラグレベルの 4 個のレベル、異常終了処理診断部分では、異常終了コード 240-0C6 基本情報レベル、異常終了コード 240-0C7 基本情報レベル、異常終了コード 240-0C7 検索レベルの 4 個のレベルがある。また全体の制御を行う全体制御レベルがある。異常終了コード 240-0C7 検索レベルは、1 個のダイナミック・ノードを定義してありそれ以外のレベルはそれぞれ 1 個のスタティックノードを定義してある。アトリビュートは全部で 91 個ある。以下に各レベルの概要を述べる。

a. DO ループ基本情報レベル (アトリビュート数 9 個)

DO ループの多重度、ループ長、ベクトル長、配列宣言の大きさなどのベクトル化・チューニングのための基本情報を保持する。DO ループの多重度だけは端末から利用者によって入力される。

b. ベクトル化質問レベル (アトリビュート数 37 個)

質問に対して端末から入力される利用者の答えを保持する。

c. ベクトル化診断制御レベル (アトリビュート数 2 個)

ベクトル化診断の KS の制御を行うフラグを保持する。

d. 推奨技法フラグレベル（アトリビュート数 42個）

診断を行う KS で設定される推奨技法のフラグを保持する。

e. 異常終了コード 240-0C6 基本情報レベル（アトリビュート数 7 個）

異常終了コード C6 の現象に対する質問の答えおよび対策のフラグを保持する。

f. 異常終了コード 240-0C7 基本情報レベル（アトリビュート数 17個）

異常終了コード C7 の原因となったサブルーチン名やデータ名、検索の制御フラグを保持する。

g. 異常終了コード 240-0C7 検索レベル（アトリビュート数 18個）

検索中のデータの名前、タイプ、サブルーチン名、下位ルーチン名等を保持する。異常終了の原因の候補であるデータ毎に 1 つずつダイナミック・ノードが作られ結合される。

h. 全体制御レベル（アトリビュート数 1 個）

初期メニューの表示の制御フラグを保持する。

(3) その他

・推論エンジン

ESHELL で提供している推論エンジンを使用した。探索は深さ優先である。

・LISP 定義関数

主に利用者からの入力サポート・画面制御を行うマン・マシン・インターフェースを向上する関数群とループ長をキーとしたソート等の推論を行う上で必要となる関数群がある。利用者の入力は軽減化され、なるべく YES または NO または数字で行えるようになっている。

3.4 実行例

(1) ベクトル化・チューニング診断の実行例

2 章で示した Fig. 2.1 を例に Ces-VP を実行した例を Fig. 3.6 で示す。Ces-VP の質問には Fig. 2.1 のプログラムを見て、YES または NO または数字等で入力するようになっている。この問題の場合の診断状況を示す。まず DO 文のベクトル化表示について質問がされる。この DO 文の横のベクトル化表示は、ベクトル・コンパイラが output するもので対象の DO ループのすべての文がベクトル化されている場合は V、全てベクトル化されていない場合は S、混在している場合は M になる。この場合 S が表示されているので NO を入力する。次にベクトル化コンパイラから出力されるベクトル化メッセージの番号 228 を入力する。そして DO ループの多重度や配列に関する質問の後に IF 文の有無が質問されるが、ここでは無いので NO を入力する。YES を入力するとさらに IF 文の入れ子等の質問がされる。そして次に 228 番のメッセージに関する質問で外部手続き SUBX の構造が尋ねられる。SUBX はステートメント数が多いという条件があったので NO を入力する。このような必要な質問が終わると診断結果としてベクトル化メッセージの意味とその対処方法つまりベクトル化技法が推奨される。この場合外部手続きを引用している部分とその他の部分に DO ループを分割し、その外部手続きの中に DO ループを持ち込むことを推奨している。親ルーチンで DO 文や CONTINUE 文、SUBX の CALL 文以外に実行文がある場合は、DO ループを CALL 文の前後で分割できるかを検討する必要があるが、この場合無いので親ルーチンの DO ループをそのまま SUBX に持ち込み引数を配列で渡すように修正すればよい。

(2) 異常終了処理診断の実行例

① 異常終了コード 240-0C6 の場合

Fig. 2.6 で示すようなプログラムをFORTRAN77/VPでコンパイルしVP-100で実行するとトレースバック情報に示すようにコード 240-0C6 で異常終了する。これをCes-VPで診断した例をFig. 3.7に示す。まず、選択メニューで異常終了の原因調査の“2”を入力する。そしてスカラー処理で正常終了しているかが確認される。Fig. 3.7 のプログラムは正常に終了するので“YES”を入力する。次に異常終了コードを“C6”を入力する。そして*VOCL LOOP, REPEAT文に関する質問がされる。トレースバック情報で異常終了したのはサブルーチン SUB1 の ISN 5 番であるが、DO 10 のループには*VOCL LOOP, REPEAT 文の指定がないので“NO”を入力する。次にエラーで起きたルーチン SUB1 の引数について質問がされる。この場合 ISN 4 番に実数型 8 バイトの配列 A があるので“YES”を入力する。次にエラーの起きたルーチンを呼び出しているルーチン MAIN で、SUB1 を呼び出す際の実引数 IA, IB が 8 バイトの変数と結合されているかを質問する。X と IA, IB が結合されているので“YES”を入力する。すると C6 の異常終了の原因と診断結果が出力される。診断結果は、実引数の 4 バイトの配列が 8 バイト境界にないので修正するように指示する。

② 異常終了コード 240-0C7 の場合

Fig. 2.8 で示すようなプログラムをFORTRAN77/VPでコンパイルしVP-100で実行するとコード C7 で異常終了する。これをCes-VPで診断した例をFig. 3.8に示す。メニューの選択やスカラー処理での正常終了の確認は前述の 0C6 の場合と同様である。コード“C7”を入力する。その後異常終了したデータについて質問がされる。トレースバック情報を参考に入力すると異常終了したルーチン名は SUB2, データ名が RA, データタイプが実数型 8 バイトである。するとデータ RA がどのように定義されているかがメニュー形式で質問される。RA は引数であり上位ルーチン MAIN で定義されているので“6”を入力する。次に SUB2 の中で RA と EQUAI VALENCE 文で結合されている変数名の存在が質問される。この場合は存在しないので“NO”を入力する。次に RA と引数で結合されているデータの定義を行う。この場合上位ルーチン名は MAIN, 実引数は IA で整数型 4 バイトを入力する。上位ルーチンは MAIN プログラム 1 つのみなので定義はここで終了する。次に MAIN のデータ IA の定義を入力する。この場合 ISN 6 番で 0 でない値を入力しているので“3”を入力する。また IA と他のデータの結合があるかが質問される。ないので“NO”を入力する。すると異常終了 C7 の原因が MAIN においてデータ IA が整数型 4 バイトなのに実数型 8 バイトのデータ RA と結合していることを指摘し宣言タイプの修正を指示する。

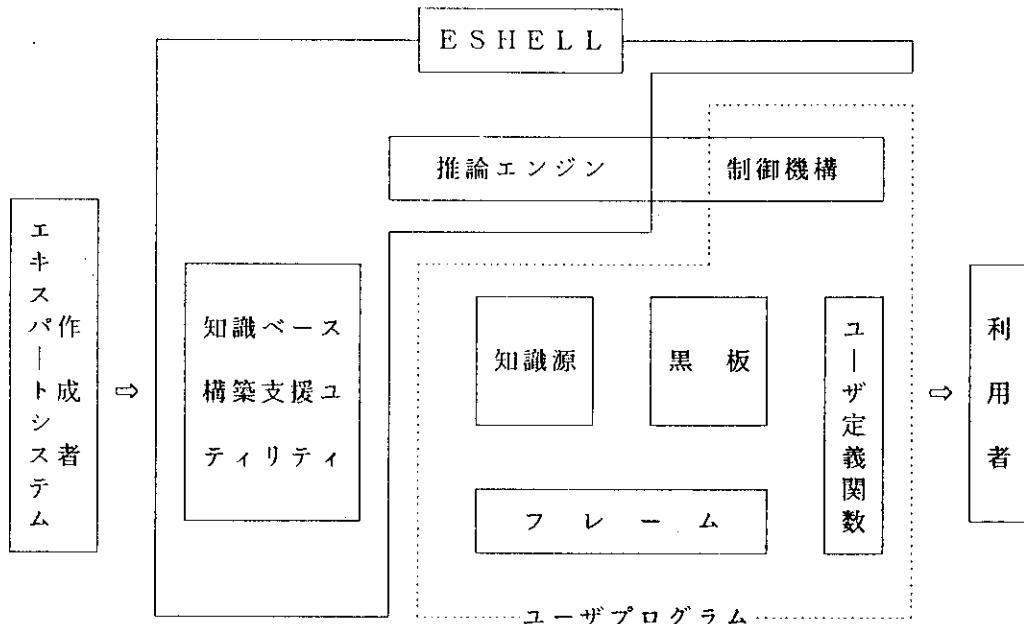


Fig. 3.1 System configuration of expert shell EHELL.

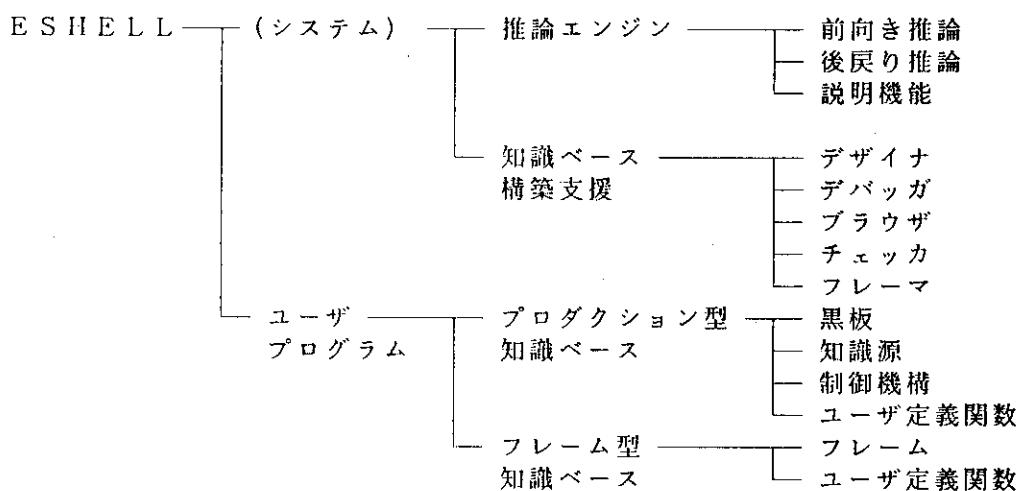
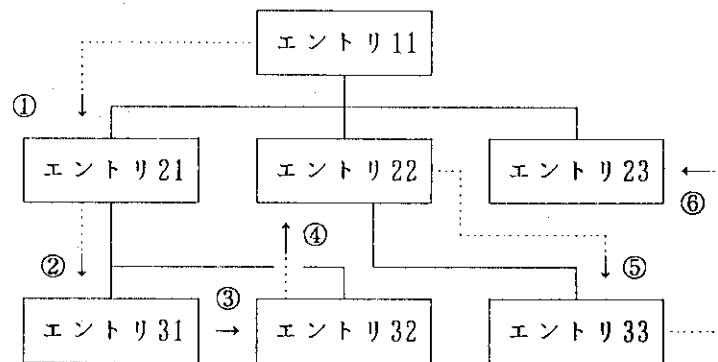


Fig. 3.2 The relation between the components of EHELL.

(a) depth-first-search



(b) width-first-search

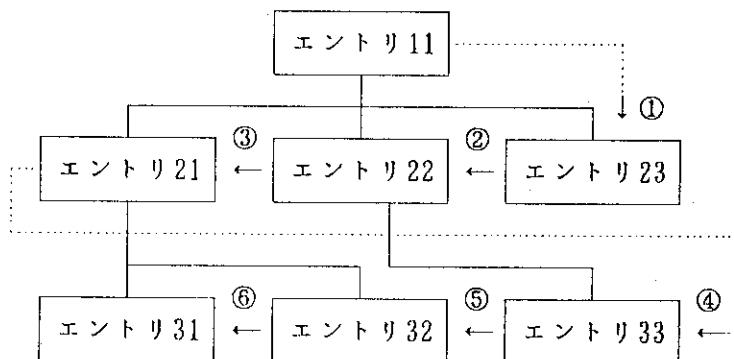


Fig. 3.3 The order of rule search.

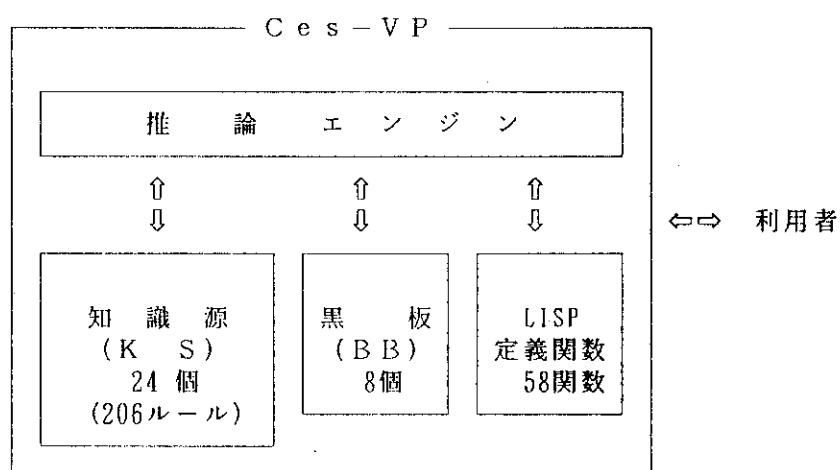


Fig. 3.4 System configuration of Ces-VP.

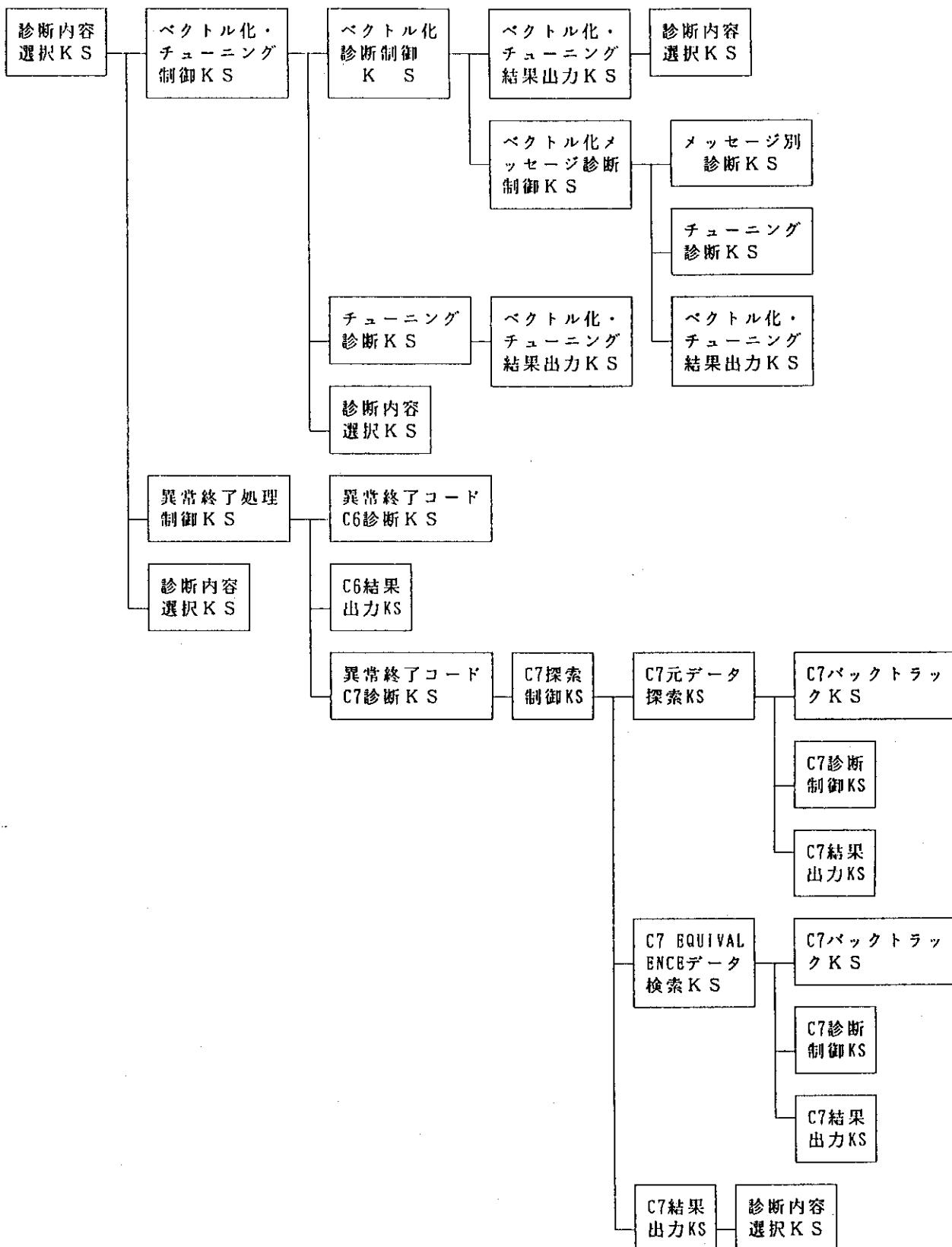


Fig. 3.5 Knowledge source structure of Ces-VP.

```
*****  
**  
** ベクトル計算機用プログラム開発支援システム  
**  
** C e s - V P  
**  
** ( CONSULTING EXPERT SYSTEM FOR VECTOR PROGRAMMING )  
** これは、日本原子力研究所殿と富士通㈱との共同開発品です。  
*****  
診断内容の番号を選択して下さい。  
1. ベクトル化およびチューニング  
2. アベンドの原因調査  
3. 終了  
(1/2/3/HELP) ==> 1
```

使用方法は、以下の通りです。

1. 会話型ベクトライザのバッチ出力を用意して下さい。
 2. 一番コストの高いプログラム単位（サブルーチンなど）から、コスト順に以下の手順を繰り返してください。
 3. 一番コストの高いD O ループからコスト順に、あるいは、上から順に以下の手順を繰り返してください。
 4. D O ループに対するベクトル化メッセージ及びループ構造を調べて、質問に答えてください。
- 尚、質問には、特に指定の無い限り、“YES”又は“NO”で答えてください。

ENTER KEYを押してください。

D O 文のベクトル化表示は、Vになっていますか？

VかMの場合は、YES

Sの場合は、NO

初期メニューにもどる場合は、QUIT

実行を終了する場合は、END

を入力してください。

(YES/NO/QUIT/END)

==> NO

ベクトル化メッセージ (J N D n n n) が出力されていますか？

(YES/NO)

==> YES

ベクトル化メッセージ (J N D n n n) の番号は、いくつですか？

n n n をすべて入力してください。

データを終了する時はENDを入力して下さい。

==> 228

==> END

Fig. 3.6 An example of vectorizing program in Fig. 2.1 by using Ces-VP.

このDOループは、何重ループですか？

==> 1

内側のDO文から、DOループ変数名、繰り返し回数、繰り返し回数の確定度を順番に入力してください。

内側から1番目のDO文の定義を開始します。

DOループ変数名 ==> I

繰り返し回数（1以上の整数値） ==> 100

DOループ変数の増分値（0以外の整数値） ==> 1

繰り返し回数の確定度（確定値なら1、予想値なら0） ==> 0

1組のデータの定義を終了しました。

ベクトル化されているDOループは、内側から何番目ですか？
(ベクトル化されていない場合は、1を入力してください。)

==> 1

使用されている主な配列の次元はいくつですか？

==> 1

配列の第1番目の添字に書かれている変数名を定義します。

DOループ変数ならその名前、配列ならDIMENSION、定数ならその値
(DOループ内で変わらない変数などは、予想値でも可) を入力して下さい。

==> 1

DOループの中にIF文がありますか？

(YES/NO)

==> NO

メッセージJND-228の対象となった外部手続きの構造が単純であり、且つステップ数が少ないのでですか？

(YES/NO)

==> NO

* * * * *

* * ベクトル化診断結果 *

* * * * *

次に示すような点について検討してください。

<メッセージ番号 JND-228 の意味 >

外部手続(CALI文、関数引用)は、ベクトル化できない。

* * メッセージ番号JND-228に対する処置 *

外部手続を利用している部分とその他の部分にDOループを分割し
その外部手続の中にDOループを持ち込むことを検討して下さい。
診断はすべて終了しました。ENTERキーを押してください。

Fig. 3.6 (continued)

* ベクトル計算機用プログラム開発支援システム
* Ces-VP
* (CONSULTING EXPERT SYSTEM FOR VECTOR PROGRAMMING)
* これは、日本原子力研究所殿と富士通㈱との共同開発品です。

診断内容の番号を選択して下さい。

- ベクトル化およびチューニング
 - アベンドの原因調査
 - 終了

(1/2/3/HELP) => 2

使用方法は、以下の通りです。

1. 異常終了した時のエラートレースおよびソースプログラムを用意して下さい。

尚、質問には、特に指定の無い限り、「YES」又は「NO」で答えてください。

スカラ-処理をして正常終了しましたか?
(YES/NO)

==> YES

アベンドコード 240 の詳細コードを入力して下さい。
(C6 OR C7)

$\Rightarrow C_6$

エラーの起きた D O ループの D O 文の直前には、
* V O C L L O O P, R E P E A T (n) 文があり、
かつ n が D O ループの繰り返し回数を超えていませんか?
(YES/NO)
==> NO

— 1 —

エラーの起きたルーチンの仮引数の中で、実数型8バイトの配列または変数があり、かつその実引数が実数または整数型4バイトの配列として宣言されていますか？
(YES/NO)

YES

エラーの起きたルーチンを呼び出しているルーチンでは、呼び出す際の実引数が実数型 8 バイトの配列または変数と EQUIVALENCE 文で結合されていますか？
(YES/NO)

YES

異常終了の原因

アベンドコード 0 C 6 は、ベクトル処理をする倍精度データが
8 バイト境界上にない場合に出力されます。

実引数の 4 バイトの配列を含む E Q U I V A L E N C E 文を正しくうなづかせて下さい。

ENTER KEYを押してください

Fig. 3.7 An example of fixing program in Fig. 2.6 by using Ces-VP.

```
*****
* * ベクトル計算機用プログラム開発支援システム *
* * Ces-VP *
* * ( CONSULTING EXPERT SYSTEM FOR VECTOR PROGRAMMING ) *
* * これは、日本原子力研究所殿と富士通㈱との共同開発品です。 *
*****
```

診断内容の番号を選択して下さい。

1. ベクトル化およびチューニング
2. アベンドの原因調査
3. 終了

(1/2/3/HELP) ==> 2

使用方法は、以下の通りです。

1. 異常終了した時のエラートレースおよびソースプログラムを用意して下さい。

尚、質問には、特に指定の無い限り、“YES”又は“NO”で答えてください。

スカラー処理をして正常終了しましたか？
(YES/NO)

==> YES

アベンドコード240の詳細コードを入力して下さい。
(C6 OR C7)
==> C7

異常終了したルーチンの名前を入力して下さい。==> SUB2
異常終了したベクトル演算を行なっているデータの名前を入力して下さい。==> RA
異常終了したデータのタイプを入力して下さい。(R*4,R*8)==> R*8

ルーチン SUB2 において RA は、
エラーを起こす直前には、どのような定義がされていますか？
次の中から選択して番号を入力して下さい。

1. A・Z形の書式つきREAD文や書式なしREAD文で読みこんでいるか
または文字型の変数・定数、HEXデータを代入している。
2. 固定・浮動小数点で値が0であるデータを代入している。
3. 固定・浮動小数点で値が0でないデータを代入している。
4. COMMON変数であり、他のルーチンで定義されている。
5. 間接的な変数・配列要素で定義されていて且つそれが
COMMON変数であり、他のルーチンで定義されている。
6. 仮引数であり、上位ルーチンで定義されている。
7. 間接的な変数・配列要素で定義されていて且つそれが
仮引数であり、上位ルーチンで定義されている。
8. 下位ルーチンに実引数として渡され、定義されている。
9. 未定義になっている。

==> 6

Fig. 3.8 An example of fixing program in Fig. 2.8 by using Ces-VP.

SUB2 の中で RA と他の変数・配列が
E Q U I V A L E N C E 文で結合され、かつ RA
より新らしく定義されていますか？
(YES/NO)
==> NO

ルーチン名 SUB2 において
仮引数名 RA (タイプ R*8) を含む
仮引数の名前、仮引数のタイプ、上位ルーチンの名前、
上位ルーチンで書かれている実引数の名前、実引数のタイプ
を順番に入力して下さい。タイプは、整数型 2・4 バイト ==> I*2・I*4,
実数型 4・8 バイト ==> R*4・R*8、文字型 ==> CHR を入力して下さい。

データ NO.1 の定義を開始します。
上位ルーチンの名前 ==> MAIN
実引数の名前 ==> IA
実引数のタイプ (I*2, I*4, R*4, R*8, CHR) ==> I*4
データが定義されました。次のデータを定義しますか？(YES OR NO) ==> NO

ルーチン MAIN において IA は、
ルーチン SUB2 に渡される直前には、どのような定義がされていますか？
次の中から選択して番号を入力して下さい。

1. A・Z 形の書式つき READ 文や書式なし READ 文で読みこんでいるか
または文字型の変数・定数、HEXデータを代入している。
 2. 固定・浮動小数点で値が 0 であるデータを代入している。
 3. 固定・浮動小数点で値が 0 でないデータを代入している。
 4. COMMON 変数であり、他のルーチンで定義されている。
 5. 間接的な変数・配列要素で定義されていて且つそれが COMMON 変数であり、他のルーチンで定義されている。
 6. 仮引数であり、上位ルーチンで定義されている。
 7. 間接的な変数・配列要素で定義されていて且つそれが
仮引数であり、上位ルーチンで定義されている。
 8. 下位ルーチンに実引数として渡され、定義されている。
 9. 未定義になっている。
- ==> 3

MAIN の中で IA と他の変数・配列が
E Q U I V A L E N C E 文で結合され、かつ IA
より新らしく定義されていますか？
(YES/NO)
==> NO

異常終了の原因

アベンドコード 0 C 7 は、ベクトル演算をするデータが、
非正規化データの場合に出力されます。

```
* * * * * * * * * * * * * * *
*   診断結果   *
* * * * * * * * * * * * * * *
```

ルーチン MAIN において
データ IA が宣言タイプ I*4 であるのに対し、下位ルーチン SUB2 における
引数および COMMON 変数 RA の宣言タイプが R*8 であるためと考えられます。

この宣言タイプが一致するように修正してください。

ENTER KEY を押してください。

Fig. 3.8 (continued)

4. Ces-VPの評価と議論

Ces-VPを試作後、その性能を評価し問題点を考察するために、原研計算センターで原子力コードのプログラム開発、改良に従事する10人を対象にアンケート調査を行った。

4.1 アンケート要領

アンケートの対象者は、FORTRAN経験2年～8年、ベクトル化経験0.1～1.5年程度である。ベクトル化に関しては、初級者が2人であり、残り8人は中級から上級者の部類に入りCes-VPの対象レベルよりは高い。アンケートは、これまでに原研でベクトル化したプログラムから特徴的な14個のDOループを対象者にベクトル化してもらい、その時の性能、工数、マン・マシン・インターフェース（以下MMIと呼ぶ）について調査する。まず、対象者10人を経験年数等で平均的になるように5人づつAグループとBグループの2つのグループに分ける。そして問1～問14までのサンプル・プログラムを1問づつ両方のグループにベクトル化してもらった。ここで、Ces-VPの使用効果を計り、かつ、両方のグループの実力差を縮めるために、Aグループは、奇数番号の問題はCes-VPを使用してベクトル化し、偶数番号の問題はCes-VPを使用しないでベクトル化した、一方、Bグループは逆に奇数番号の問題はCes-VPを使用しないでベクトル化し、偶数番号の問題はCes-VPを使用してベクトル化した。そして、各問題をベクトル化したら次のような3つの点について回答をしてもらった。

(a) 性能

- Ces-VPが出力したベクトル化技法
- 対象者が選んだベクトル化技法
- ベクトル化したプログラムの速度

(b) 工数

- ベクトル化に要した工数（プログラムを実行した時の計算機の待ち時間を除く）

ベクトル化は、プログラムを見てどのようなベクトル化技法を選択するかを決定し、プログラムを修正して、FORTUNE¹¹⁾で処理時間の見積りを行うか実際にVP-100を使用して処理時間を計る。さらにその結果が不満ならばベクトル化方法を再考するという繰り返しを行う。これらの作業時間を0時間～1時間はレベル1、1時間～2時間はレベル2、2時間～3時間はレベル3、3時間以上はレベル4としてベクトル化工数と定義した。

(c) システムのMMI

- 質問の意味、Ces-VPの推奨技法の意味等に関する不明確な点・改善意見等のコメント

4.2 アンケート結果

問題別の調査の狙いおよび最適技法をTable 4.1に示す。アンケート調査のまとめをTable 4.2

に示す。調査に使用したソース・プログラムと、それを専門家が最適技法でベクトル化したソース・プログラムをFig. 4.1～Fig. 4.14に示す。各問題別の調査結果はTable 4.3～Table 4.16に示す。各図中の網かけの欄は最適技法を示している。

このアンケートのまとめでは、以下に示すような量を定義している。まず、専門家がベクトル化をするのに利用した技法を最適技法と呼ぶ。その最適技法に関する量として、最適技法推奨率、最適技法使用者数および最適技法使用率を以下のように定義する。

$$\text{最適技法推奨率} = \frac{\text{最適技法の中で Ces-VP が推奨した技法の総数}}{\text{最適技法の数} \times 5 \text{ 人}} \times 100$$

$$\text{最適技法使用者数} = \text{最適技法だけに注目し対象者の使用技法の数を総和}$$

$$\text{最適技法使用率} = \frac{\text{最適技法使用者数の総和}}{\text{最適技法の数} \times 5 \text{ 人}} \times 100$$

(実際に対象者がどの程度最適技法を使用しているかを表している)

次に、Ces-VP が推奨した技法の使用者数と使用率を以下のように定義する。

$$\text{推奨技法使用者数} = \text{Ces-VP の推奨した技法で実際対象者が使用した数}$$

$$\text{推奨技法使用率} = \frac{\text{推奨技法使用者数の総和}}{\text{Ces-VP の推奨数}} \times 100$$

(Ces-VP の技法がどの程度実際に使用されているかを表している)

Ces-VP を使用した場合としない場合のベクトル化したプログラムの処理効率向上率と作業工数短縮率とを以下のように定義する。

$$\text{Ces-VP による速度向上率} = \frac{\text{Ces-VP 使用側の平均倍率}}{\text{Ces-VP 未使用側の平均倍率}}$$

$$\text{Ces-VP による工数短縮率} = \frac{\text{Ces-VP 未使用側の平均工数}}{\text{Ces-VP 使用側の平均工数}}$$

アンケート結果の評価を行う前に、次のような四つの問題があり、これらを排除する努力は行っているが、完全に影響を除外はできていないことを指摘しておく。

- Ces-VP の使用側と未使用側のグループ分け

Ces-VP の使用効果を調査するため同じ問題をCes-VP を使用するグループと使用しないグループに分けた。このグループ間の実力がFORTRAN 経験とベクトル化経験によって同じになるようにしたが、個人的な実力がまちまちであるため実力を均等にする事が難しかった。

- ベクトル化工数の算出

プログラムの実行がバッチ処理なため結果が出てくるまでの計算機の待ち時間が存在する。またその待ち時間の間に次の問題を考えている場合があり正確な時間を引き出すのが困難であった。またCes-VPを使用している場合、慣れれば1分～5分程度であるがCes-VPでの応答時間も含まれる。

- 学習効果

Ces-VPを使用しない問題を先に回答してもらったが、Ces-VPを使ってみると後の問題ほどベクトル化経験が学習効果で増していった。

- 工数不足の効果

このアンケート調査は、対象者の仕事に割り込む形で短時間で行われたため個人的に許容される工数が違った。

4.3 議論

コンパイラの自動ベクトル化機能は、解析範囲が翻訳単位であるためにサブルーチンや関数などの手続き内にあるDOループに関しては非常に有効である。しかし、原子力コードのように、繰り返し処理が複数のサブルーチンにわたるような複雑なプログラムに関してはコンパイラの自動ベクトル化機能の有効性は制限されている。そのような問題にたいしてCes-VPシステムを使うことによって、簡単な問題に対してベクトル化や最適化をする事ができるようになった。アンケート調査をまとめたTable 4.2から判るように、Ces-VPの使用側と未使用側で大きな速度向上は見られなかった。これは、ほとんどの調査対象者がベクトル化知識を豊富にもっているためであると考えられる。また、Ces-VPは、Table 4.3～4.16に示すようにTable 4.1の問題の狙いに示す技法をほぼ推奨している。ただし、今回のプロトタイプ・システムを試作し、また、アンケート調査を行った結果として、今後Ces-VPシステムをベクトル化支援のための実用エキスパート・システムとするためには次のような二つの大きな問題がある。

- ルールの充実

- ソースプログラムの解析結果の利用

最初のルールの充実については、今回のプロトタイプ・システムが教育資料に基づいてベクトル化・チューニングのルールを抽出したために、アンケートに使用した程度のプログラムでもルールに不充分なものがあった。たとえば、現在16個のベクトル化メッセージに対応できるようになっているが、それ以外のメッセージに対しては対応できない。また、問6, 7, 9, 12に現れるように多重DOループのDO文間にスカラ変数がある場合には、スカラ変数を取り除くような最適化を行ってから、DOループの分割・入れ替え・一重化などのチューニングを行わなければならない。ところが、現状のルールでは、スカラ最適化を行うためのルールがないために、ベクトル・チューニングを行えない。

次のソースプログラムの解析結果の利用というのは、現状のプロトタイプでは考えているDOループが多重DOループかどうか等のプログラムの分析をCes-VPの利用者に行ってもらっている。そのため、Ces-VPの推奨技法として、チューニングのための修正方法が比較的画一的なプログラム修正の自由度が小さい技法（問2, 3, 4, 8, 12, 14の推奨技法：多次元

配列の第1添字の変更, アンローリング, 配列のアクセス方向の変更, 文字型変数の型変換, DOループの入れ換え, 最内DOループの書き下し, ベクトル化メッセージに対応する作業配列の作成, 作業配列の変数化, *VOCL LOOP, REPEAT文の挿入, *VOCL LOOP, SCALAR文の挿入, *VOCL LOOP, NOVREC文の挿入, インライン展開, 下位ルーチンへのDOループの持ち込み) では速度の向上が見られる。しかし, 修正の方法がいくつもあり現在の推奨方法で効率的な修正をおこなうためにはかなり知識がなければならないプログラム修正の自由度が大きい技法(問1, 5, 6, 7, 9, 10, 11, 13の推奨技法 DOループの1重化, DOループの分割, 作業配列の作成)では, 速度向上効果が見られない。

例えば, 問5, 8, 9, 11で現れているように現在のCes-VPでは, ベクトル化メッセージやベクトル・コンパイラの出力するベクトル化表示, IF文, ステートメント数からDOループの分割を推奨しているが, 具体的にソースプログラムの解析をしている訳ではないので分割の場所を指示することはきれない。また, ソースプログラムを解析してベクトル化後にどのようなタイミングで処理がされるのかという評価のルールを持たないので, 複数の分割方法が考えられる場合に, どの方法が最良かを指示することができない。

また, ソースプログラムを解析した結果を利用すれば, 現在は使用者が入力しているプログラム構造に関するデータの入力量が減り利用し易いシステムになると考えられる。

以上のような事から今後実用化システムとするためには次のことを考える必要がある。

(1) 推奨技法の使用例の表示

Ces-VPシステムが技法を推奨した時に, その技法を使用したプログラムを例示する。例えばDOループの1重化の場合は, 配列を1次元化するための方法やインデックス計算の計算式の例などをプログラムで例示する。

(2) プログラム解析ツールおよびベクトル化効果評価ツールとの結合

ソースプログラムを解析して, その結果をエキスパート・システムで利用することは二つの意味で重要である。一つは, プログラムの構造を利用者に入力してもらう必要がなくなり利用者の負担が減ることである。他の意味は, ベクトル化の診断をした後で, 具体的にどのように修正すれば良いかを指示することである。

ベクトル化評価ツールとの結合では, 複数のベクトル化・チューニングの可能性がある場合に, ベクトル化効果を評価して, 効果の大きい方法を指示することができる。

(3) ルールの追加

エキスパート・システムで一般的に問題となるように, 新しいルールを自動的にとり入れていくような学習機能を持たせる工夫をする。

Table 4.1 Optimal technique for problems of questionnaire

問題番号	調査の狙いおよび最適技法
1 狙い	<p>: 2重DOループの内側のループが既にベクトル化されていて、かつ計算が1ステートメントしかないのでチューニング能力が期待される。</p> <p>最適技法：最速技法は、DOループの1重化（技法①）と*VOCL LOOP, REPEAT文の挿入（技法④）である。</p> <p>最高倍率：2.5倍</p> <p>その他：配列の全要素をアクセスしているわけではないので、現段階のCes-VPではDOループの1重化を推奨できない。</p>
2 狙い	<p>: 配列の添字が降順に変化するような計算を持つ1重DOループ。配列のアクセス方向による速度の違いの知識が期待される。</p> <p>最適技法：最速技法は、DO文の変数の増分値を負から正に変え、配列のアクセス方向が昇順になるような修正、さらに*VOCL LOOP, REPEAT文の挿入</p> <p>最高倍率：1.3倍</p>
3 狙い	<p>: 一般的には、ベクトル化できない文字型変数でも代入のみで使用されている場合はベクトル化できる。比較的特殊であるが原子力コードではたまに見られるので出題した。特殊な技法なため知らない人はベクトル化を断念せざるえない。</p> <p>最適技法：文字型変数を引数またはイクリバレンス文によって実数型または整数型変数と結合し、さらにスカラーの最適化を行う。</p> <p>最高倍率：6.7倍</p>
4 狙い	<p>: 最内DOループの書き下しおよびDOループの入れ換えの知識が期待される。3重DOループの最内ループのループ長は3で総和計算を行っている。</p> <p>最適技法：最内ループを書き下して、Iの関するループを内側に入れ換え、*VOCL LOOP, REPEAT文の挿入を行う。</p> <p>最高倍率：18倍</p>
5 狙い	<p>: 2重DOループでループ長10程度でベクトル化されている。DOループの入れ換えによるベクトル長の増加および配列のアクセス効率を考慮したDOループの分割が期待される。</p> <p>最適技法：DOループの入れ換え・分割・1重化である。</p> <p>最高倍率：12倍</p>
6 狙い	<p>: 2重DOループで比較的複雑な総和計算をする問題である。DOループの分割とそれに伴う作業配列の作成を期待しているが、かなりの試行錯誤が予想される。</p> <p>最適技法：1次元の作業配列の作成とDOループの分割でベクトル化し総和計算の部分となるべく簡略化するようにする。</p> <p>最高倍率：2.7倍</p>
7 狙い	<p>: 2重DOループの内側DOループ変数に関して総和計算し、外側DOループ変数を添字に持つ配列に格納する。ベクトル化していてもスカラーの最適化によって早くなる場合のチューニング能力が期待される。</p> <p>最適技法：無駄な作業変数を削除した後DOループを分割し、入れ換えてアンローリングを行い、配列のアクセスを効率良くする。</p> <p>最高倍率：2.8倍</p> <p>その他：Ces-VPは、スカラーの最適化を考慮していないので、この問題の場合にはDOループの入れ換えの選択条件が阻害され、推奨しない。</p>

Table 4.1 (continued)

8	狙い	2重DOループの中にさらに1重DOループが2つあり総和計算を行っている。外側のループで定義され内側のループで使用しているベクトル化阻害変数の配列化およびDOループの分割の能力が期待される。 最適技法：ベクトル化阻害変数を配列化し、DOループを分割してさらに入れ換えてアンローリング・多次元配列の第1添字の変更・*VOCL LOOP, REPEAT文の挿入などを行う。 最高倍率：2.2倍 その他：Ces-VPは、DOループの分割の後の状態を把握できないためDOループの入れ換えは推奨できない。
9	狙い	実行ステートメント60は2重DOループだが外側のDO文と内側のDO文の間にIF文や多くの変数が定義され内側のDOループで使用している。問8と同様にベクトル化阻害変数の削除とDOループの分割の分け目の判断能力が問われる。 最適技法：ベクトル化阻害変数の配列化とそれに伴うDOループの分割及びスカラー最適化やDOループの入れ換え等で、さらに多次元配列の第1添字の変更や*VOCL LOOP, REPEAT文の挿入を行う。 最高倍率：2.8倍
10	狙い	2重DOループの内側DOループのループ変数に関して回帰参照であるためベクトル化できない。回帰参照の回避のためのDOループの入れ換えを知っているかが問われる。 最適技法：内側のループと外側のループを入れ換えて内側のループ変数だけに依存する計算を別ループにDOループの分割し作業配列を使ってアクセスし、さらに多次元配列の第1添字の変更を行う。 最高倍率：15倍
11	狙い	2重DOループで内側に1重DOループが2つあり、その間にIF-GOTO文がある。この内側のループはベクトル長が短く、外側のループ長は長いのでDOループの分割・入れ替えの能力が問われる。 最適技法：DOループの分割をし、入れ換えて外側のループでベクトル化する方法でDOループの分割するループ間での変数を配列化し、さらに*VOCL LOOP, REPEAT文の挿入も行う。 最高倍率：29倍
12	狙い	2重DOループの内側のループで、主な計算を行っているサブルーチンを呼び出している。このサブルーチンはDOループがなくベクトル化されない。DOループ内に外部手続きがある場合のベクトル化技法を知っているかが問われる。 最適技法：呼び出しているサブルーチンを内側ループに展開し、さらにDOループの分割をして無駄な計算を除去して*VOCL LOOP, REPEAT文の挿入を行う。 最高倍率：10倍
13	狙い	3重DOループで各ループ変数から新しい添字変数を計算し、条件によって0か1を代入する問題。回帰参照除去の知識が期待される。 最適技法：まず0を代入するDOループと1を代入するループに分割しその後スカラーの最適を行う。 最高倍率：5.8倍 その他：回帰参照の可能性を示唆するベクトル化メッセージJND-235が outputされるがCes-VPでは考慮していない。また*VOCL LOOP, (条件) 文も考慮に入っていないが*VOCL LOOP, NOVREC文で代用で推奨される。
14	狙い	2重DOループで外側と内側のDO文の間と内側ループの最初で添字の計算を行い、内側ループで配列をアクセスする。添字の計算部分の最適化と回帰参照の除去を期待している。 最適技法：外側と内側のDO文の間を最適化し、*VOCL LOOP, NOVREC文の挿入で回帰参照のないことを指示し、さらに*VOCL LOOP, REPEAT文の挿入を行う。 最高倍率：2.7倍 その他：Ces-VPでは、DO文とDO文の間に文がある場合、それが削除不可能であることを仮定している。したがって、DOループの入れ換え・分割・1重化に影響している。

Table 4.2 Summary of results of questionnaire

問題番号	最高倍率	最適技法推奨率	倍率	工数	最適技法使用率	推奨技法使用率	速度向上率	工数短縮率	ベクトル化自由度
1	2.5	30	2.0 (1.4)	1.2 (3.0)	40 (10)	47	0.89	1.5	大
2	1.3	80	1.2 (1.2)	1.0 (1.0)	70 (50)	62	1.0	1.0	小
3	6.7	100	6.6 (4.3)	1.0 (2.2)	70 (30)	100	1.5	2.2	小
4	18	89	17 (17)	1.0 (1.0)	67 (60)	38	1.0	1.0	小
5	12	70	6.5 (9.3)	2.2 (1.2)	45 (50)	46	0.70	0.86	大
6	2.7	4	1.7 (2.2)	2.8 (3.8)	28 (48)	25	0.77	1.4	大
7	2.8	25	1.2 (2.2)	1.8 (2.2)	30 (90)	27	0.56	1.2	大
8	2.2	33	1.7 (1.6)	2.4 (2.6)	53 (47)	62	1.1	1.1	小
9	2.8	50	1.3 (1.5)	3.0 (3.6)	40 (43)	38	0.87	1.2	大
10	15	75	8.5 (8.9)	1.6 (1.8)	40 (45)	39	0.94	1.1	大
11	29	29	10 (23)	1.6 (1.8)	50 (70)	50	0.43	1.1	大
12	10	45	7.5 (8.1)	2.0 (2.4)	32 (36)	65	0.93	1.2	小
13	5.8	20	4.9 (4.7)	1.4 (2.0)	27 (27)	44	1.0	1.4	大
14	2.7	20	1.4 (1.4)	1.2 (1.4)	27 (27)	45	1.0	1.2	小
平均	8.1	48	5.1 (6.2)	1.8 (2.1)	44 (45)	49	0.82	1.2	

() 内はCes-VP未使用側のデータ

Table 4.3 The summary of answer for problem no.1

技法番号*	⑩	⑥	⑨	⑦	③							
Ces-VP 使用	Ces-VPの推奨数		5	3	4	3					倍 率	工数
	対象者の使用技法			○	○						1.3	1
	"		○	○							1.3	3
	"		○								1.1	1
	"			○	○						1.3	1
	"			○							1.2	4
Ces-VP 未使用	最適技法使用者数	0	4									
	推奨技法使用者数	2	3	2	0							
Ces-VP 未使用	対象者の使用技法	○									2.4	3
	"										1.0	4
	"		○								1.2	3
	"		○								1.4	1
	"										1.0	4
最適技法使用者数		1	0									

Table 4.4 The summary of answer for problem no.2

技法番号*	⑧	⑨	⑦							
Ces-VP 使用	Ces-VPの推奨数	5	3	5					倍率	工数
	対象者の使用技法	○	○						1.3	1
	"	○							1.2	1
	"	○							1.2	1
	"	○	○						1.2	1
	"	○	○						1.3	1
Ces-VP 未使用	最適技法使用者数	5	2							
	推奨技法使用者数	5	2	1						
Ces-VP 未使用	対象者の使用技法	○							1.2	1
	"	○							1.2	1
	"	○							1.2	1
	"	○							1.2	1
	"	○							1.2	1
Ces-VP 未使用	最適技法使用者数	5	0							

Table 4.5 The summary of answer for problem no.3

aスカラー最適化

Table 4.6 The summary of answer for problem no.4

技法番号*	④	②	⑨	⑦	⑥	⑩	⑮	㉑		
Ces-VPの推奨数	5	5	3	5	4	3	3	1		
C e s — V P 使 用	対象者の使用技法	○	○	○					18	1
	"	○							11	1
	"	○	○						18	1
	"	○	○		○				18	1
	"	○	○						18	1
	最適技法使用者数	5	4	1						
C e s — V P 未 使 用	推奨技法使用者数	5	4	1	1	0	0	0		
	対象者の使用技法	○	○						18	1
	"	○	○						18	1
	"	○	○						18	1
	"	○							11	1
	"	○	○						18	1
最適技法使用者数		5	4	0						

Table 4.7 The summary of answer for problem no.5

技法番号*		b	②	④	⑨	⑩	⑦	⑥	⑤	⑫	c		
Ces-VPの推奨数		2	5	3	0	4	5	4	1	3	1	倍率	工数
Ces-VP 使用	対象者の使用技法		○	○			○					5.5	1
	"			○			○					1.4	3
	"		○	○								9.9	1
	"		○	○	○		○					10	1
	"		○					○				5.6	1
最適技法使用者数		2	4	3	0								
推奨技法使用者数		2	4	3	0	0	2	1	1	0	0		
Ces-VP 未使用	対象者の使用技法		○	○								10	1
	"		○	○	○							11	2
	"		○	○								9.9	1
	"		○	○								9.9	1
	"		○	○								5.6	1
最適技法使用者数		4	5	0	1								

b 配列の1次元化

c 作業配列の変数化

Table 4.8 The summary of answer for problem no.6

技法番号*		d	③	e	f	⑩	⑤	⑦	⑥	g	④	⑨		
Ces-VPの推奨数		0	0	-	3	1	4	5	5	1	4	1	倍率	
Ces-VP 使用	対象者の使用技法		○	○	○			○					2.4	1
	"		○		○		○						1.1	4
	"		○		○	○	○						1.5	2
	"						○		○				1.7	4
	"		○		○	○							1.9	3
最適技法使用者数		1	4	1	1									
推奨技法使用者数				3	1	2	3	0	1	0	0			
Ces-VP 未使用	対象者の使用技法		○	○	○								2.1	4
	"		○	○		○	○	○					2.5	4
	"		○	○	○					○			2.6	4
	"		○	○	○								2.7	3
	"												1.0	4
最適技法使用者数		4	4	3	1									

d 1次元作業配列の作成
e スカラー最適化f 2次元作業配列の作成
g 作業配列の変数化

Table 4.9 The summary of answer for problem no.7

技法番号*		③	④	h	⑥	⑨	i	j	⑦		
C e s — V P 使 用	Ces-VPの推奨数	0	0	—	6	3	0	2	5		
	対象者の使用技法			○						1.7	4
	"	○	○	○						1.4	1
	"			○	○	○	○			1.0	2
	"				○					1.0	1
	"	○								1.1	1
最適技法使用者数		2	1	2	1						
推奨技法使用者数				1	2		1	0			
C e s — V P 未 使 用	対象者の使用技法	○	○	○	○					2.8	2
	"	○	○	○						1.7	3
	"	○	○	○	○					2.7	1
	"	○	○	○			○			1.6	2
	"	○	○	○	○					2.1	3
	最適技法使用者数	5	5	4	3						

h スカラー最適化
i 作業配列の生成

j 作業配列の変数化

Table 4-10 The summary of answer for problem no.8

技法番号*	k	③	④	⑦	⑫	⑯	⑯				
Ces-VPの推奨数	4	1	0	4	0	4	0				
	対象者の使用技法	◎	○	○	○					倍率	工数
	"	◎	○	○	○					2.0	1
	"	◎	○	○	○					1.9	3
	"	◎	○	○	○					1.6	4
	"	◎	○	○	○					1.8	2
	"	◎	○	○	○					1.3	2
最適技法使用者数	5	5	3	3	0	0					
推奨技法使用者数	4	1		3	0						
Ces-VP未使用	対象者の使用技法	◎		○						1.4	4
	"	◎	○	○	○					1.7	2
	"	○	○	○						1.6	2
	"	◎	○	○						1.7	1
	"	◎	○	○						1.6	4
最適技法使用者数	5	4	5	0	0	0					

k 作業配列の作成

Table 4.11 The summary of answer for problem no.9

技法番号*		①	③	m	④	⑦	⑨	⑥	⑤			
C e s — V P 使 用	Ces-VPの推奨数	5	3	—	0	7	3	4	2			倍率 工数
	対象者の使用技法	◎										1.2 4
	"	◎	○									1.4 3
	"	◎	○	○	○	○						1.3 3
	"	◎		○								1.4 1
	"	◎										1.0 4
最適技法使用者数		5	3	2	1	0	1					
推奨技法使用者数		5	2		0	1	0	0				
C e s — V P 未 使 用	対象者の使用技法	◎	○	○	○	○						2.7 2
	"	◎	○									1.7 4
	"	○	○									1.2 4
	"			○	○							1.0 4
	"	○	○									1.0 4
	最適技法使用者数	4	4	2	2	1	0					

1 作業配列の作成

mスカラーの最適化

Table 4.12 The summary of answer for problem no.10

技法番号*		③	④	n	⑦	○	⑥	⑤				
C e s — V P 使 用	Ces-VPの推奨数	5	5	0	5	0	4	4				倍率 工数
	対象者の使用技法	○	○	○	○							12 1
	"	○	○									14 2
	"	○	○									14 3
	"					○						1.0 1
	"					○						1.0 1
最適技法使用者数		3	3	1	1							
推奨技法使用者数		3	3		1	0	2					
C e s — V P 未 使 用	対象者の使用技法	○										1.2 3
	"	○	○			○						14 2
	"	○	○	○								14 2
	"	○	○									14 1
	"	○										1.2 1
	最適技法使用者数	5	3	1	0							

n 作業配列の作成

○ 作業配列の変数化

Table 4.13 The summary of answer for problem no.11

技法番号*		p	③	④	⑫	⑦	⑬	⑨	⑥			
C e s — V P 使 用	Ces-VPの推奨数	2	3	0	0	4	3	2	4			倍率 工数
	対象者の使用技法	○	○	○								4.0 2
	"	○	○	○								24 2
	"	○	○	○								20 2
	"				○	○						3.4 1
	"		○		○							1.0 1
	最適技法使用者数	3	4	3			0					
C e s — V P 未 使 用	推奨技法使用者数	2	3			1	1	0	0			
	対象者の使用技法	○	○	○								27 1
	"	○	○	○								23 3
	"	○	○	○								20 1
	"	○	○	○								20 2
	"	○	○	○								26 2
最適技法使用者数		4	5	5			0					

p 作業配列の配列

Table 4.14 The summary of answer for problem no.12

技法番号*		⑪	⑫	⑬	q	τ	⑩	⑨	⑥			
C e s — V P 使 用	Ces-VPの推奨数	3	2	3	5	—	0	1	3			倍率 工数
	対象者の使用技法	○		○	○	○	○					7.8 1
	"	○			○							7.4 3
	"		○		○							7.4 2
	"	○										7.5 1
	"		○	○								7.3 3
	最適技法使用者数	3	2	2		1		0				
C e s — V P 未 使 用	推奨技法使用者数	3	2	2	3	—	0	0				
	対象者の使用技法	○		○	○							7.6 4
	"		○	○	○							7.4 3
	"	○			○							9.8 1
	"	○				○						7.8 1
	"		○		○							7.7 3
最適技法使用者数		3	2	2	2	—	0					

q 作業配列の作成

τスカラ-最適化

Table 4.15 The summary of answer for problem no.13

技法番号*	③	s	⑤	t	⑨	⑩	⑥	⑦	④		
Ces-VPの推奨数	1	—	4	0	2	0	4	4	1		倍率
			○		○					5.6	2
	"	○								4.4	1
	"		○							4.7	2
	"		○							5.6	1
	"	○	○							4.4	1
最適技法使用者数	2	0			2						
推奨技法使用者数	1		4		2		0	0	0		
Ces-VP未使用	対象者の使用技法			○		○				4.6	2
	"		○			○				4.3	4
	"	○	○							5.3	2
	"	○								4.4	1
	"		○	○						4.7	1
最適技法使用者数	2	2			0						

Sスカラ-最適化

t * VOCAL LOOP, (条件) 文の插入

Table 4.16 The summary of answer for problem no.14

技法番号*	⑤	u	③	v	w	⑨	⑦	④	⑥		
Ces-VPの推奨数	2	4	4	—	1	3	1	5		倍率	工数
C e s — V P 使 用	対象者の使用技法	○	○	○						1.5	1
	"	○	○	○						1.5	2
	"		○	○						1.8	1
	"		○				○			1.1	1
	"	○								1.1	1
	最適技法使用者数	3	—	1		0	0				
C e s — V P 未 使 用	推奨技法使用者数	2		3	3	0	0	1	0	0	
	対象者の使用技法	○		○						1.7	3
	"	○								1.1	1
	"	○								1.1	1
	"	○								1.1	1
	"		○	○						1.8	1
最適技法使用者数		4	0		0	0					

u * VOCAL LOOP, (条件) 文の挿入

Wスカラ-最適化

v 作業配列の作成

```

PROGRAM EX01
IMPLICIT REAL*8 (A-H,O-Z)
C-> DECLARE,DEFINE DATA
PARAMETER (IMAX=10,JMAX=15)
REAL*8 A(IMAX,JMAX),B(IMAX,JMAX)
DATA S/1.D0/
NO = 0
DO 10 J=1,JMAX
DO 10 I=1,IMAX
NO = NO + 1
A(I,J)=0.D0
10 B(I,J)=REAL(NO)
C-----
DATA IOVER/1000/
OVERH = 0.D0
DO 1 I=1,IOVER
CALL CLOCK(STIME,2,2)
OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
C-----
C-> ORIGINAL VERSION
DO 2 I=1,100000
CALL SUB01(A,B,IMAX,JMAX,S)
2 CONTINUE
C   WRITE(6,*) A(1,1),A(10,10),A(IMAX,JMAX)
C-----
CALL CLOCK(TTIME,2,2)
S = (TTIME-STIME-OVERH)/10**6
WRITE(6,1000) S
1000 FORMAT(1H , 'SUB01 TIME =',1PD14.3)
STOP
END
SUBROUTINE SUB01(A,B,IMAX,JMAX,S)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 A(IMAX,JMAX),B(IMAX,JMAX)
DO 10 I=1,IMAX-1
DO 11 J=1,JMAX
A(I,J)=S*B(I,J) + A(I+1,J)
11 CONTINUE
10 CONTINUE
RETURN
END

```

Fig. 4.1 (a) Original source program of problem no.1

```

SUBROUTINE SUB01V(AA,BB,A%,S,IMAX,JMAX)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 AA(IMAX*JMAX+1),BB(IMAX*JMAX),A%(JMAX),A(IMAX,JMAX)
DO 5 J=1,JMAX
A%(J)= A(IMAX,J)
5 CONTINUE
DO 10 IJ=1,IMAX*JMAX
AA(IJ)=S*BB(IJ) + AA(IJ+1)
10 CONTINUE
DO 20 J=1,JMAX
A(IMAX,J)= A%(J)
20 CONTINUE
RETURN
END

```

Fig. 4.1 (b) Vectorized program of problem no.1 by expert

```

PROGRAM EX02
IMPLICIT REAL*8 (A-H,O-Z)
C-> DECLARE,DEFINE DATA
PARAMETER (IMAX=100,JMAX=100)
REAL*8 A(IMAX,JMAX),B(IMAX)
DATA L/100/
DATA S/1.D0/
DATA A,B/10100*0.D0/
DO 10 I=1,L
10   A(I,L) = I
C-----
      DATA IOVER/1000/
      OVERH = 0.D0
      DO 1 I=1,IOVER
      CALL CLOCK(STIME,2,2)
      OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
C-----
C-> ORIGINAL VERSION
DO 2 I=1,1000000
CALL SUB02(A,B,L,IMAX,S)
2 CONTINUE
C-----
      CALL CLOCK(TTIME,2,2)
      S = (TTIME-STIME-OVERH)/10**6
      WRITE(6,1000) S
1000 FORMAT(1H ,SUB02 TIME =',1PD14.3)
      STOP
      END
      SUBROUTINE SUB02(A,B,L,IMAX,S)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 A(IMAX,1),B(IMAX)
DO 10 I=L,1,-1
      B(I) = B(I) + S *A(I,L)
10 CONTINUE
      RETURN
      END

```

Fig. 4.2 (a) Original source program of problem no.2

```

SUBROUTINE SUB02V(A,B,L,IMAX,S)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 A(IMAX,1),B(IMAX)
DO 10 I=1,L
      B(I) = B(I) + S *A(I,L)
10 CONTINUE
      RETURN
      END

```

Fig. 4.2 (b) Vectorized program of problem no.2 by expert

```

PROGRAM EX03
IMPLICIT REAL*8 (A-H,O-Z)
C-> DECLARE,DEFINE DATA
PARAMETER (IMAX=100)
CHARACTER*8 A(IMAX),B(IMAX)
DATA IL/100/
DATA A/100*'ABCDEFGH'/
C-----
DATA IOVER/1000/
OVERH = 0.D0
DO 1 I=1,IOVER
CALL CLOCK(STIME,2,2)
OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
C-----
C-> ORIGINAL VERSION
DO 2 I=1,100000
CALL SUB03(A,B,IL,IMAX)
2 CONTINUE
C-----
CALL CLOCK(TTIME,2,2) *
S = (TTIME-STIME-OVERH)/10**6
WRITE(6,1000) S
1000 FORMAT(1H ,SUB03 TIME =',1PD15.3)
STOP
END
SUBROUTINE SUB03(A,B,IL,IMAX)
IMPLICIT REAL*8 (A-H,O-Z)
CHARACTER*8 A(IMAX),B(IMAX)
DO 10 I=1,IL
    B(I)=A(I)
10 CONTINUE
RETURN
END

```

Fig. 4.3 (a) Original source program of problem no.3

```

SUBROUTINE SUB03V(A,B,IL,IMAX)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 A(IMAX),B(IMAX)
DO 10 I=1,IL
    B(I)=A(I)
10 CONTINUE
RETURN
END

```

Fig. 4.3 (b) Vectorized program of problem no.3 by expert

```

PROGRAM EX04
IMPLICIT REAL*8 (A-H,O-Z)
C-> DECLARE,DEFINE DATA
PARAMETER (IMAX=100,JMAX=10)
REAL*8 A(IMAX,JMAX,3),B(IMAX),C(JMAX)
REAL*8 D(3)
DATA IL,JL/100,10/
DATA A,B,C/3110*1.D0/
C-----
DATA IOVER/1000/
OVERH = 0.D0
DO 1 I=1,IOVER
CALL CLOCK(STIME,2,2)
OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
C-----
C-> ORIGINAL VERSION
DO 2 I=1,10000
CALL SUB04(A,B,C,D,IMAX,JMAX,IL,JL)
2 CONTINUE
C-----
CALL CLOCK(TTIME,2,2)
S = (TTIME-STIME-OVERH)/10**6
WRITE(6,1000) S
1000 FORMAT(1H , 'SUB04    TIME =',1PD15.3)
STOP
END
SUBROUTINE SUB04(A,B,C,D,IMAX,JMAX,IL,JL)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 A(IMAX,JMAX,3),B(IMAX),C(JMAX)
REAL*8 D(3)
DO 10 I=1,IL
DO 10 J=1,JL
DO 10 K=1,3
A(I,J,K)=A(I,J,K)+B(I)*C(J)*D(K)
10 CONTINUE
RETURN
END

```

Fig. 4.4 (a) Original source program of problem no.4

```

SUBROUTINE SUB04V(A,B,C,D,IMAX,JMAX,IL,JL)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 A(IMAX,JMAX,3),B(IMAX),C(JMAX)
REAL*8 D(3)
DO 10 J=1,JL
DO 10 I=1,IL
A(I,J,1)=A(I,J,1)+B(I)*C(J)*D(1)
A(I,J,2)=A(I,J,2)+B(I)*C(J)*D(2)
A(I,J,3)=A(I,J,3)+B(I)*C(J)*D(3)
10 CONTINUE
RETURN
END

```

Fig. 4.4 (b) Vectorized program of problem no.4 by expert

```

PROGRAM EX05
IMPLICIT REAL*8 (A-H,O-Z)
C-> DECLARE,DEFINE DATA
PARAMETER (IMAX=100,JMAX=10)
REAL*8 A(IMAX,JMAX),B(IMAX),C(JMAX)
REAL*8 D(IMAX,JMAX),E(IMAX)
INTEGER*4 L(IMAX)
DATA IL,JL/100,10/
DATA A,B,C/1110*1.0D/
DO 100 I=1,IMAX
100 L(I) = MOD(I,JMAX)+1
C-----
DATA IOVER/1000/
OVERH = 0.D0
DO 1 I=1,IOVER
CALL CLOCK(STIME,2,2)
OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
C-----
C-> ORIGINAL VERSION
DO 2 I=1,10000
CALL SUB05(A,B,C,D,E,IMAX,JMAX,IL,JL,L)
2 CONTINUE
C-----
CALL CLOCK(TTIME,2,2)
S = (TTIME-STIME-OVERH)/10**6
WRITE(6,1000) S
1000 FORMAT(1H , 'SUB05 TIME =',1PD15.3)
STOP
END
SUBROUTINE SUB05(A,B,C,D,E,IMAX,JMAX,IL,JL,L)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 A(IMAX,JMAX),B(IMAX),C(JMAX)
REAL*8 D(IMAX,JMAX),E(IMAX)
INTEGER*4 L(IMAX)
DO 10 I=1,IL
DO 10 J=1,JL
D(I,J)=A(I,J)+B(I)*C(J)
E(I) = A(I,L(I))+ C(L(I))
10 CONTINUE
RETURN
END

```

Fig. 4.5 (a) Original source program of problem no.5

```
SUBROUTINE SUB05V(A,B,C,D,E,IMAX,JMAX,IL,JL,L)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8   A(1),B(1),C(1)
REAL*8   D(IMAX,1),E(1)
INTEGER*4 L(1)
DO 10 J=1,JL
*VOCL LOOP,REPEAT(100)
DO 20 I=1,IL
      K = (J-1) * IMAX + I
      D(I,J)=A(K)+B(I)*C(J)
20 CONTINUE
10 CONTINUE
*VOCL LOOP,REPEAT(10)
DO 30 I=1,JL
      K =(L(I)-1) * IMAX + I
      E(I) = A(K)+ C(L(I))
30 CONTINUE
RETURN
END
```

Fig. 4.5 (b) Vectorized program of problem no.5 by expert

```

PROGRAM EX06
IMPLICIT REAL*8 (A-H,O-Z)
C-> DECLARE,DEFINE DATA
PARAMETER (I1MAX=100,JRI=100)
REAL*8 A(I1MAX,JRI),B(JRI)
LOGICAL*4 MASK(I1MAX),MASK2(I1MAX)
DATA I1/100/
DATA A,B/10100*1.D0/
DATA FAC/1.D0/
DATA MASK/100*.TRUE./
DATA MASK2/100*.TRUE./
C-----
DATA IOVER/1000/
OVERH = 0.D0
DO 1 I=1,IOVER
CALL CLOCK(STIME,2,2)
OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
C-----
C-> ORIGINAL VERSION
DO 2 I=1,1000
CALL SUB06(A,B,I1,I1MAX,JRI,FAC,SUM,MASK,MASK2)
2 CONTINUE
C-----
CALL CLOCK(TTIME,2,2)
S = (TTIME-STIME-OVERH)/10**6
WRITE(6,1000) S
1000 FORMAT(1H ,SUB06 TIME =',1PD15.3)
STOP
END
SUBROUTINE SUB06(A,B,I1,I1MAX,JRI,FAC,SUM,MASK,MASK2)
IMPLICIT REAL*8 (A-H,O-Z)
REAL *8 A(I1MAX,JRI),B(JRI)
LOGICAL*4 MASK(I1MAX),MASK2(I1MAX)
SUM = 0.D0
DO 1 I=2,I1-1
DO 2 J=1,JRI
IFC MASK(I).OR.A(I,J).EQ.0.D0) THEN
IFC MASK2(I) ) THEN
SUM=SUM+A(I,J)*(B(I)**2+B(I+1)**2*FAC)
ELSE
SUM=SUM+A(I,J)*(B(I)**2+B(I-1)**2*FAC)
END IF
IFC SUM.LT.0.D0 ) THEN
SUM=ABS(SUM)
END IF
ELSE
A(I,J) = ABS(A(I,J))
END IF
2 CONTINUE
1 CONTINUE
RETURN
END

```

Fig. 4.6 (a) Original source program of problem no.6

```

SUBROUTINE SUB06V(A,B,I1,I1MAX,JRI,FAC,SUM,MASK,MASK2)
IMPLICIT REAL*8 (A-H,O-Z)
REAL    *8 A(I1MAX,JRI),B(JRI)
REAL    *8 WORK(100),WORK2(100)
LOGICAL*4 MASK(I1MAX),MASK2(I1MAX)
SUM = 0.0D0
*VOCL LOOP,REPEAT(100)
DO 4000 I=2,I1-1
  IF( MASK2(I) ) THEN
    WORK(I) = B(I)**2+B(I+1)**2*FAC
  ELSE
    WORK(I) = B(I)**2+B(I-1)**2*FAC
  END IF
4000 CONTINUE
DO 3 I=2,I1-1
*VOCL LOOP,REPEAT(100)
DO 4100 J = 1, JRI
  WORK2(J) = A(I,J)*WORK(I)
4100 CONTINUE
IF( MASK(I) ) THEN
*VOCL LOOP,SCALAR
  DO 2 J=1,JRI
    SUM=ABS(SUM+WORK2(J))
  2 CONTINUE
ELSE
*VOCL LOOP,REPEAT(100)
  DO 4 J=1,JRI
    A(I,J) = ABS(A(I,J))
  4 CONTINUE
END IF
3 CONTINUE
RETURN
END

```

Fig. 4.6 (b) Vectorized program of problem no.6 by expert

```

PROGRAM EX07
PARAMETER (NXMAX=100)
IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 X(NXMAX),Y(NXMAX),Z(NXMAX),C(NXMAX,NXMAX)
DATA X,Y/200*0.D0/
DATA Z/100*1.D0/
DATA C/10000*1.D0/
DATA NX/100/
DATA IOVER/1000/
OVERH = 0.D0
DO 1 I=1,IOVER
CALL CLOCK(STIME,2,2)
OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
DO 2 I=1,10000
CALL SUB07(X,Y,Z,C,NXMAX,NX)
2 CONTINUE
CALL CLOCK(TTIME,2,2)
S = (TTIME-STIME-OVERH)/10**6
WRITE(6,1000) S
1000 FORMAT(1H , 'SUB07 TIME =',1PD15.3)
STOP
END
SUBROUTINE SUB07(X,Y,Z,C,NXMAX,NX)
REAL*8 X(NXMAX),Y(NXMAX),Z(NXMAX),C(NXMAX,1),P
DO 1 I=1,NX
P=0.D0
DO 2 J=1,NX
2 P = P + C(I,J)*Z(J)
Y(I) = P
1 X(I) = P
RETURN
END

```

Fig. 4.7 (a) Original source program of problem no.7

```

SUBROUTINE SUB07V(X,Y,Z,C,NXMAX,NX)
REAL*8 X(NXMAX),Y(NXMAX),Z(NXMAX),C(NXMAX,1)
*VOCL LOOP,REPEAT(100)
DO 1 I=1,NX
  Y(I) = 0.0D0
1 CONTINUE
IF(MOD(NX,2).EQ.0) THEN
  DO 2 J=1,NX,2
*VOCL LOOP,REPEAT(100)
  DO 2 I=1,NX
    Y(I) = Y(I) + C(I,J)*Z(J) + C(I,J+1)*Z(J+1)
2 CONTINUE
ELSE
  DO 20 J=1,NX-1,2
*VOCL LOOP,REPEAT(100)
  DO 20 I=1,NX
    Y(I) = Y(I) + C(I,J)*Z(J) + C(I,J+1)*Z(J+1)
20 CONTINUE
*VOCL LOOP,REPEAT(100)
  DO 21 I=1,NX
    Y(I) = Y(I) + C(I,NX)*Z(NX)
21 CONTINUE
END IF
*VOCL LOOP,REPEAT(100)
DO 3 I=1,NX
  X(I) = Y(I)
3 CONTINUE
RETURN
END

```

Fig. 4.7 (b) Vectorized program of problem no.7 by expert

```

PROGRAM EX08
PARAMETER (NMAX=100, MMAX=100)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION AR(NMAX,MMAX),AI(NMAX,MMAX),ZR(NMAX,MMAX),ZI(NMAX,MMAX)
DATA N,M/2*100/
DO 10000 I=1,N
DO 10000 J=1,M
IF(I.EQ.J) THEN
AI(I,J)=1.D2
AR(I,J)=2.D2
ZI(I,J)=3.D2
ZR(I,J)=4.D2
ELSE
AI(I,J)=1.DO
AR(I,J)=2.DO
ZI(I,J)=3.DO
ZR(I,J)=4.DO
END IF
10000 CONTINUE
DATA IOVER/1000/
OVERH = 0.DO
DO 1 I=1,IOVER
CALL CLOCK(STIME,2,2)
OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
DO 2 I=1,100
CALL SUB08(N,NMAX,M,MMAX,AR,AI,ZR,ZI)
2 CONTINUE
CALL CLOCK(TTIME,2,2)
S = (TTIME-STIME-OVERH)/10**6
WRITE(6,1000) S
1000 FORMAT(1H , 'SUB08 TIME =',1PD15.3)
STOP
END
SUBROUTINE SUB08(N,NMAX,M,MMAX,AR,AI,ZR,ZI)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION AR(NMAX,MMAX),AI(NMAX,MMAX),ZR(NMAX,MMAX),ZI(NMAX,MMAX)
DO 140 I=2,N
L=I-1
H=AI(I,I)
IF(H.EQ.0.DO) GO TO 140
DO 130 J=1,M
S=0.DO
SI=0.DO
DO 110 K=1,L
S=S+AR(I,K)*ZR(K,J)-AI(I,K)*ZI(K,J)
SI=SI+AR(I,K)*ZI(K,J)+AI(I,K)*ZR(K,J)
110 CONTINUE
C ***** DOUBLE DIVISIONS AVOID POSSIBLE UNDERFLOW *****
S = (S/H)/H
SI= (SI/H)/H
DO 120 K=1,L
ZR(K,J)=ZR(K,J)-S*AR(I,K)-SI*AI(I,K)
ZI(K,J)=ZI(K,J)-SI*AR(I,K)-S*AI(I,K)
120 CONTINUE
130 CONTINUE
140 CONTINUE
RETURN
END

```

Fig. 4.8 (a) Original source program of problem no.8

```

SUBROUTINE SUB08V(N,NMAX,M,MMAX,AR,AI,ZR,ZI,SR¥,SI¥)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION AR(NMAX+1,MMAX),AI(NMAX+1,MMAX)
DIMENSION ZR(NMAX+1,MMAX),ZI(NMAX+1,MMAX)
DIMENSION SR¥(MMAX),SI¥(MMAX)
DO 1140 I=2,N
L=I-1
H=AI(I,I)
IF(H.NE.0.D0) THEN
*VOCL LOOP,REPEAT(100)
DO 1130 J=1,M
SR¥(J)=0.D0
SI¥(J)=0.D0
1130 CONTINUE
DO 1110 K=1,L-1,2
*VOCL LOOP,REPEAT(100)
DO 1110 J=1,N
SR¥(J)=SR¥(J)+AR(I,K)*ZR(K,J)-AI(I,K)*ZI(K,J)
& +AR(I,K+1)*ZR(K+1,J)-AI(I,K+1)*ZI(K+1,J)
SI¥(J)=SI¥(J)+AR(I,K)*ZI(K,J)+AI(I,K)*ZR(K,J)
& +AR(I,K+1)*ZI(K+1,J)+AI(I,K+1)*ZR(K+1,J)
1110 CONTINUE
IF(MOD(L-1,2).EQ.0) THEN
*VOCL LOOP,REPEAT(100)
DO 1115 J=1,N
SR¥(J)=SR¥(J)+AR(I,L)*ZR(L,J)-AI(I,L)*ZI(L,J)
SI¥(J)=SI¥(J)+AR(I,L)*ZI(L,J)+AI(I,L)*ZR(L,J)
1115 CONTINUE
END IF
C ***** DOUBLE DIVISIONS AVOID POSSIBLE UNDERFLOW *****
*VOCL LOOP,REPEAT(100)
DO 1111 J=1,N
SR¥(J)=(SR¥(J)/H)/H
SI¥(J)=(SI¥(J)/H)/H
1111 CONTINUE
DO 1120 J=1,M-1,2
*VOCL LOOP,REPEAT(100)
DO 1120 K=1,L
ZR(K,J)=ZR(K,J)-SR¥(J)*AR(I,K)-SI¥(J)*AI(I,K)
ZR(K,J+1)=ZR(K,J+1)-SR¥(J+1)*AR(I,K)-SI¥(J+1)*AI(I,K)
ZI(K,J)=ZI(K,J)-SI¥(J)*AR(I,K)-SR¥(J)*AI(I,K)
ZI(K,J+1)=ZI(K,J+1)-SI¥(J+1)*AR(I,K)-SR¥(J+1)*AI(I,K)
1120 CONTINUE
IF(MOD(M-1,2).EQ.0) THEN
*VOCL LOOP,REPEAT(100)
DO 1125 K=1,L
ZR(K,M)=ZR(K,M)-SR¥(M)*AR(I,K)-SI¥(M)*AI(I,K)
ZI(K,M)=ZI(K,M)-SI¥(M)*AR(I,K)-SR¥(M)*AI(I,K)
1125 CONTINUE
END IF
END IF
1140 CONTINUE
RETURN
END

```

Fig. 4.8 (b) Vectorized program of problem no.8 by expert

```

PROGRAM EX09
PARAMETER (NTMAX=100,NE2MAX=100)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION EM(NE2MAX),S(4),SNT(NE2MAX),CNT(NE2MAX)
DIMENSION FTP(NE2MAX,NTMAX),FTS(NE2MAX,NTMAX),
.          FTD(NE2MAX,NTMAX),FTF(NE2MAX,NTMAX),
.          CTP(NE2MAX,NTMAX),CTS(NE2MAX,NTMAX),
.          CTD(NE2MAX,NTMAX),CTF(NE2MAX,NTMAX),
.          STP(NE2MAX,NTMAX),STS(NE2MAX,NTMAX),
.          STD(NE2MAX,NTMAX),STF(NE2MAX,NTMAX)
DATA NE2,NT/2*100/
DATA NE1,NL/10,5/
DATA I1,I2,I3,I4/1,2,3,4/
DATA V/1.D0/
DATA EM/100*1.D0/
DATA S/0.1D0,0.2D0,3.D0,4.D0/
DATA SNT,CNT/200*0.D0/
DATA IOVER/1000/
OVERH = 0.D0
DO 1 I=1,IOVER
CALL CLOCK(STIME,2,2)
OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
DO 2 I=1,1000
CALL SUB09(NE1,NE2,NT,EM,S,SNT,CNT,FTD,FTF,FTP,FTS,
>           STD,STF,STP,STS,CTD,CTF,CTP,CTS,V,NL,
>           I1,I2,I3,I4,NE2MAX,NTMAX)
2 CONTINUE
CALL CLOCK(TTIME,2,2)
SC = (TTIME-STIME-OVERH)/10**6
WRITE(6,1000) SC
1000 FORMAT(1H ,SUB09 TIME =',1PD15.3)
STOP
END
SUBROUTINE SUB09(NE1,NE2,NT,EM,S,SNT,CNT,FTD,FTF,FTP,FTS,
>           STD,STF,STP,STS,CTD,CTF,CTP,CTS,V,NL,
>           I1,I2,I3,I4,NE2MAX,NTMAX)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION EM(NE2MAX),S(4),SNT(NE2MAX),CNT(NE2MAX)
DIMENSION FTP(NE2MAX,NTMAX),FTS(NE2MAX,NTMAX),
.          FTD(NE2MAX,NTMAX),FTF(NE2MAX,NTMAX),
.          CTP(NE2MAX,NTMAX),CTS(NE2MAX,NTMAX),
.          CTD(NE2MAX,NTMAX),CTF(NE2MAX,NTMAX),
.          STP(NE2MAX,NTMAX),STS(NE2MAX,NTMAX),
.          STD(NE2MAX,NTMAX),STF(NE2MAX,NTMAX)
C
DO 25 L=NE1,NE2
EE=EM(L)
IF(EE.LE.S(1)) GOTO 25
IF(EE.LE.S(4)) GOTO 20
IF(EE.LE.S(2)) GOTO 21
IF(EE.LE.S(3)) GOTO 22
DELTA=S(4)+S(3)-S(2)-S(1)
AM=(S(4)*S(3)-S(2)*S(1))/DELTA
D1=S(1)-AM
D2=S(2)-AM

```

Fig. 4.9 (a) Original source program of problem no.9

```

V1=V/DELTA
DEE=EE-AM
FRAC=DEE*DEE/D2/D1
CN=3.DO*V1*(1.DO-FRAC)
SN=3.DO*V1*DEE-V1*(D1+D2+DEE*FRAC)
GO TO 23
22 DELTA=S(4)+S(3)-S(2)-S(1)
AM=(S(4)*S(3)-S(2)*S(1))/DELTA
CUPD D4=S(4)-AM
D1=S(4)-AM
C/*
V1=V/DELTA
DEE=EE-S(1)
FRAC=DEE*DEE/D1/(S(1)-S(2))
CN=3.DO*V1*FRAC
SN=V1*FRAC*DEE
GO TO 23
21 DELTA=S(4)+S(3)-S(2)-S(1)
AM=(S(4)*S(3)-S(2)*S(1))/DELTA
D1=S(4)-AM
V1=V/DELTA
DEE=EE-S(1)
FRAC=DEE*DEE/D1/(S(1)-S(2))
CN=3.DO*V1*FRAC
SN=V1*FRAC*DEE
GO TO 23
20 SN=V
CN=0.DO
C
C      CALCULATE AND SUM PROJECT DENSITY AND NUMBER OF STATES
C
23 DO 24 KT=1,NT
GS=(FTS(I1,KT)+FTS(I2,KT)+FTS(I3,KT)+FTS(I4,KT))/4.DO
GP=(FTP(I1,KT)+FTP(I2,KT)+FTP(I3,KT)+FTP(I4,KT))/4.DO
GD=(FTD(I1,KT)+FTD(I2,KT)+FTD(I3,KT)+FTD(I4,KT))/4.DO
STS(L,KT)=STS(L,KT)+SN*GS
STP(L,KT)=STP(L,KT)+SN*GP
STD(L,KT)=STD(L,KT)+SN*GD
CTS(L,KT)=CTS(L,KT)+CN*GS
CTP(L,KT)=CTP(L,KT)+CN*GP
CTD(L,KT)=CTD(L,KT)+CN*GD
IF(NL.LT.4) GO TO 24
GF=(FTF(I1,KT)+FTF(I2,KT)+FTF(I3,KT)+FTF(I4,KT))/4.DO
STF(L,KT)=STF(L,KT)+SN*GF
CTF(L,KT)=CTF(L,KT)+CN*GF
24 CONTINUE
SNT(L)=SNT(L)+SN
CNT(L)=CNT(L)+CN
25 CONTINUE
RETURN
END

```

Fig. 4.9(a) (continued)

```

SUBROUTINE SUB09V(NE1,NE2,NT,EM,S,SNT,CNT,FTD,FTF,FTP,FTS,
>                 STD,STF,STP,STS,CTD,CTF,CTP,CTS,V,NL,
>                 I1,I2,I3,I4,NE2MAX,NTMAX,CN,SN,GS,GP,GD,GF)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION EM(NE2MAX),S(4),SNT(NE2MAX),CNT(NE2MAX),
DIMENSION FTP(NE2MAX+1,NTMAX),FTS(NE2MAX+1,NTMAX),
.           FTD(NE2MAX+1,NTMAX),FTF(NE2MAX+1,NTMAX),
.           CTP(NE2MAX+1,NTMAX),CTS(NE2MAX+1,NTMAX),
.           CTD(NE2MAX+1,NTMAX),CTF(NE2MAX+1,NTMAX),
.           STP(NE2MAX+1,NTMAX),STS(NE2MAX+1,NTMAX),
.           STD(NE2MAX+1,NTMAX),STF(NE2MAX+1,NTMAX)
DIMENSION CN(1),SN(1),
.           GS(1),GP(1),
.           GD(1),GF(1)
C
DELTA = S(4) + S(3) - S(2) - S(1)
AM   = ( S(4)*S(3) - S(2)*S(1) )/DELTA
D1   = S(1) - AM
D2   = S(2) - AM
D21  = D2*D1
D4   = ( S(4) - AM )*( S(1) - S(2) )
V1   = V/DELTA
C
*VOCL LOOP,REPEAT(100)
DO 4000 L = NE1, NE2
  IF(EM(L).LE.S(1)) THEN
    SN(L) = 0.D0
    CN(L) = 0.D0
  ELSE IF(EM(L).LE.S(4)) THEN
    SN(L) = V
    CN(L) = 0.D0
  ELSE IF( (EM(L).LE.S(2)) .OR. (EM(L).LE.S(3)) ) THEN
    DEE = EM(L) - S(1)
    FRAC = DEE*DEE/D4
    CN(L) = 3.D0*V1*FRAC
    SN(L) = V1*FRAC*DEE
  ELSE
    DEE = EM(L) - AM
    FRAC = DEE*DEE/D21
    CN(L) = 3.D0*V1*(1.D0-FRAC)
    SN(L) = 3.D0*V1*DEE-V1*(D1+D2+DEE*FRAC)
  END IF
  SNT(L) = SNT(L) + SN(L)
  CNT(L) = CNT(L) + CN(L)
4000 CONTINUE
C
*VOCL LOOP,REPEAT(100)
DO 4100 KT = 1, NT
  GS(KT) = ( FTS(I1,KT) + FTS(I2,KT) + FTS(I3,KT) + FTS(I4,KT) )
1      * 0.25D0
1      GP(KT) = ( FTP(I1,KT) + FTP(I2,KT) + FTP(I3,KT) + FTP(I4,KT) )
1      * 0.25D0
1      GD(KT) = ( FTD(I1,KT) + FTD(I2,KT) + FTD(I3,KT) + FTD(I4,KT) )
1      * 0.25D0
1      GF(KT) = ( FTF(I1,KT) + FTF(I2,KT) + FTF(I3,KT) + FTF(I4,KT) )
1      * 0.25D0
4100 CONTINUE

```

Fig. 4.9 (b) Vectorized program of problem no.9 by expert

```

C
C      CALCULATE AND SUM PROJECT DENSITY AND NUMBER OF STATES
C
*VOCL LOOP,REPEAT(100)
  DO 4200 KT = 1, NT
*VOCL LOOP,REPEAT(100)
  DO 4300 L = NE1, NE2
    STS(L,KT) = STS(L,KT) + SN(L)*GS(KT)
    STP(L,KT) = STP(L,KT) + SN(L)*GP(KT)
    STD(L,KT) = STD(L,KT) + SN(L)*GD(KT)
    CTS(L,KT) = CTS(L,KT) + CN(L)*GS(KT)
    CTP(L,KT) = CTP(L,KT) + CN(L)*GP(KT)
    CTD(L,KT) = CTD(L,KT) + CN(L)*GD(KT)
    IF( NL .GE. 4 ) THEN
      STF(L,KT) = STF(L,KT) + SN(L)*GF(KT)
      CTF(L,KT) = CTF(L,KT) + CN(L)*GF(KT)
    END IF
4300  CONTINUE
4200 CONTINUE
C
  RETURN
END

```

Fig. 4.9(b) (continued)

```

PROGRAM EX10
IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (KIAMAX=100,ISTM1=50)
DIMENSION QID(KIAMAX,ISTM1+1),ST(ISTM1+1)
DATA KIA/100/
DATA ISTP1/30/
DATA ST/51*0.D0/
DATA IOVER/1000/
OVERH = 0.D0
DO 1 I=1,IOVER
CALL CLOCK(STIME,2,2)
OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
DO 2 I=1,1000
CALL SUB10(QID,ST,ISTP1,KIA,KIAMAX,ISTM1)
2 CONTINUE
CALL CLOCK(TTIME,2,2)
S = (TTIME-STIME-OVERH)/10**6
WRITE(6,1000) S
1000 FORMAT(1H , 'SUB10 TIME =',1PD15.3)
STOP
END
SUBROUTINE SUB10(QID,ST,ISTP1,KIA,KIAMAX,ISTM1)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION QID(KIAMAX,1),ST(ISTM1+1)
DO 166 I = 1,KIA
QID(I,ISTP1) = 0.0
DO 165 IST=2,ISTM1
QID(I,ISTP1) = QID(I,ISTP1) + (QID(I,IST) + QID(I,IST+1))
* * (ST(IST+1)-ST(IST))* 0.5
165 CONTINUE
166 CONTINUE
RETURN
END

```

Fig. 4.10 (a) Original source program of problem no.10

```

SUBROUTINE SUB10V(QID,ST,ISTP1,KIA,KIAMAX,ISTM1,WORK)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION QID(KIAMAX,1),ST(ISTM1+1),WORK(ISTM1+1)
*VOCL LOOP,REPEAT(100)
DO 1 I = 1,KIA
QID(I,ISTP1) = 0.0
1 CONTINUE
*VOCL LOOP,REPEAT(64)
DO 2 IST=2,ISTM1
WORK(IST)= (ST(IST+1)-ST(IST))* 0.5D0
2 CONTINUE
DO 166 IST=2,ISTM1
*VOCL LOOP,REPEAT(100)
DO 166 I = 1,KIA
QID(I,ISTP1) = QID(I,ISTP1) + (QID(I,IST)+QID(I,IST+1))*WORK(IST)
166 CONTINUE
RETURN
END

```

Fig. 4.10 (b) Vectorized program of problem no.10 by expert

```

PROGRAM EX11
IMPLICIT REAL*8 (A-H,O-Z)
C-> DECLARE,DEFINE DATA
PARAMETER (KIAMAX=100,IGEMAX=50)
DIMENSION PGS(IGEMAX,KIAMAX)
DATA KIA/100/
DATA IGE2/50/
DATA PGS/2500*1.0D0,2500*0.D0/
DATA IGE1/50/
DATA V/1.0D0/
DATA IR,IOUT/3,6/
C-----
      DATA IOVER/1000/
      OVERH = 0.D0
      DO 1 I=1,IOVER
      CALL CLOCK(STIME,2,2)
      OVERH = OVERH + STIME
1 CONTINUE
      OVERH = OVERH/IOVER
      CALL CLOCK(STIME,2,2)
C-----
C-> ORIGINAL VERSION
      DO 2 I=1,10000
      CALL SUB11(KIA,KIAMAX,PGS,IGE1,IGE2,IGEMAX,V,IR,IOUT)
2 CONTINUE
C-----
      CALL CLOCK(TTIME,2,2)
      S = (TTIME-STIME-OVERH)/10**6
      WRITE(6,1000) S
      CALL CLOCK(STIME,2,2)
1000 FORMAT(1H , 'SUB11 TIME =',1PD15.3)
      STOP
      END
      SUBROUTINE SUB11(KIA,KIAMAX,PGS,IGE1,IGE2,IGEMAX,V,IR,IOUT)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PGS(IGEMAX,KIAMAX)
      . DO 520 I = 1,KIA
      IIPGS = 0
      DO 521 IGE=IGE1,IGE2
      IF(PGS(IGE,I).GE.1.0E-5) IIPGS = 1
521 CONTINUE
      IF(IIPGS.EQ.0) GOTO 520
      IF(IR .EQ.0) GOTO 523
      DO 522 IGE=IGE1,IGE2
      PGS(IGE,I) = PGS(IGE,I)/V
522 CONTINUE
523 CONTINUE
C      WRITE(IOUT,6000) PGS(IGE2,I)
6000 FORMAT(1H , 'DUMMY OUTPUT',D15.3)
520 CONTINUE
      RETURN
      END

```

Fig. 4.11 (a) Original source program of problem no.11

```

SUBROUTINE SUB11V(KIA,PGS,IGE1,IGE2,IGEMAX,KIAMAX,V,IR,WORK)
DIMENSION PGS(IGEMAX,KIAMAX), WORK(KIAMAX)

C
IF( IR .EQ. 0 )    RETURN
C
DO 4000 I = 1, KIA
  WORK(I) = 1.0D0
4000 CONTINUE
C
  V¥ = 1.0D0/V
  DO 4100 IGE = IGE1, IGE2
*VOCL LOOP,REPEAT(100)
    DO 4200 I = 1, KIA
      IF(PGS(IGE,I).GE.1.0E-5)  WORK(I) = V¥
4200  CONTINUE
4100 CONTINUE
C
  DO 4300 IGE = IGE1, IGE2
*VOCL LOOP,REPEAT(100)
    DO 4400 I = 1, KIA
      PGS(IGE,I) = PGS(IGE,I)*WORK(I)
4400  CONTINUE
4300 CONTINUE
C
  RETURN
END

```

Fig. 4.11 (b) Vectorized program of problem no.11 by expert

```

PROGRAM EX12
IMPLICIT REAL*8 (A-H,O-Z)
C-> DECLARE,DEFINE DATA
PARAMETER (NTMAX=3,JNTMAX=250)
DIMENSION SPLIT(NTMAX),NEL(NTMAX),JRIS(NTMAX),
          GAR(NTMAX,JNTMAX),RI(NTMAX,JNTMAX),
          SGX(NTMAX,JNTMAX),BGX(JNTMAX,NTMAX,NTMAX),
          COR(NTMAX,JNTMAX),GARS(JNTMAX,NTMAX,NTMAX)
DATA NT/2/
C-----
      DATA IOVER/1000/
      OVERH = 0.D0
      DO 1 I=1,IOVER
      CALL CLOCK(STIME,2,2)
      OVERH = OVERH + STIME
1 CONTINUE
      OVERH = OVERH/IOVER
      CALL INIT9(NT,SPLIT,NEL,JRIS,GAR,RI,SGX,BGX,COR,GARS,
                 NTMAX,JNTMAX)
      CALL CLOCK(STIME,2,2)
C-----
C-> ORIGINAL VERSION
      DO 2 I=1,1000
      CALL SUB12(NT,SPLIT,NEL,JRIS,GAR,RI,SGX,BGX,COR,GARS,
                 & NTMAX,JNTMAX)
2 CONTINUE
C-----
      CALL CLOCK(TTIME,2,2)
      S = (TTIME-STIME-OVERH)/10**6
      WRITE(6,1000) S
1000 FORMAT(1H , 'SUB12,SUB121 TIME =',1PD15.3)
      STOP
      END
      SUBROUTINE SUB12(NT,SPLIT,NEL,JRIS,GAR,RI,SGX,BGX,COR,GARS
                        ,NTMAX,JNTMAX)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION SPLIT(NTMAX),NEL(NTMAX),JRIS(NTMAX),
          GAR(NTMAX,JNTMAX),RI(NTMAX,JNTMAX),
          SGX(NTMAX,JNTMAX),BGX(JNTMAX,NTMAX,NTMAX),
          COR(NTMAX,JNTMAX),GARS(JNTMAX,NTMAX,NTMAX)
      DO 24 IT=1,NT
      FAC1=SPLIT(IT)/NEL(IT)/2.D0
      FAC2=0.5D0-FAC1
      JRI=JRIS(IT)
      IF(GAR(IT,JRI).LT.1.D-10) GAR(IT,JRI)=GAR(IT,JRI-1)*1.5D0+
&                                     GAR(IT,JRI-1)*0.5D0
      DO 25 IR=1,JRI
      R=RI(IT,IR)
      RCE=R*R
      RHO=GAR(IT,IR)/RCE
      RHO1=0.5D0*RHO
      RHO2=RHO1
      CALL SUB121(RHO1,RHO2,RHO,CORX,V2,EXC)
      C   WRITE(6,*) RCE,SGX(1,1),CORX
      C   SGX(IT,IR)=RCE*(SGX(IT,IR)+CORX)
      C
      C   GARS=4*PI*RADIUS**2*CHARGE DENSITY
      C   BGX=RADIUS**2*POTENTIAL

```

Fig. 4.12 (a) Original source program of problem no.12

```

C
    BGX(IR,IT,1)=SGX(IT,IR)
    BGX(IR,IT,2)=SGX(IT,IR)
    GARS(IR,IT,1)=FAC1*COR(IT,IR)+FAC2*GAR(IT,IR)
25   GARS(IR,IT,2)=GAR(IT,IR)-GARS(IR,IT,1)
24   CONTINUE
    RETURN
    END
    SUBROUTINE SUB121(RHO1,RHO2,RHO,V1,V2,EXC)
C ****
C *
C *      CALCULATES EXCHANGE CORRELATION POTENTIAL
C *
C ****
C IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CSUB12/ XCCP,XCCF,XCRP,XCRF,XALPHA,OTH,FTH,AA,BB,IXCH1
COMMON/DUMMY/EPSCP,RS,EPSCF,EPSP,EPSP,EPSP,CNY,TRX1,TRX2
C
C BARTH-HEDIN EXCHANGE CORRELATION
C J. PHYS. C5,1629(1972)
C      --
C
RSF=RS/XCRF
RSF2=RSF*RSF
RSF3=RSF2*RSF
RSP=RS/XCRP
RSP2=RSP*RSP
RSP3=RSP2*RSP
FCF=(1.D0+RSF3)*DLOG(1.D0+1.D0/RSF)+0.5D0*RSF-RSF2-OTH
FCP=(1.D0+RSP3)*DLOG(1.D0+1.D0/RSP)+0.5D0*RSP-RSP2-OTH
EPSCP=-XCCP*FCP
EPSCF=-XCCF*FCF
EPSXP=-.91633059D0/RS
CNY=5.1297628D0*(EPSCF-EPSCP)
X=RHO1/RHO
FX=(X**FTH+(1.D0-X)**FTH-AA)/BB
EXC=EPSXP+EPSCP+FX*(CNY+FTH*EPSXP)/5.1297628D0
ARS=-1.22177412D0/RS+CNY
BRS=-XCCP*DLOG(1.D0+XCRP/RS)-CNY
TRX1=(2.D0*X)**OTH
V1=ARS*TRX1+BRS
TRX2=(2.D0*RHO2/RHO)**OTH
V2=ARS*TRX2+BRS
RETURN
END
BLOCK DATA
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CSUB12/ XCCP,XCCF,XCRP,XCRF,XALPHA,OTH,FTH,AA,BB,IXCH1
COMMON/DUMMY/EPSCP,RS,EPSCF,EPSP,EPSP,CNY,TRX1,TRX2
COMMON/DUMMYV/EPSCP$,RS$,EPSCF$,EPSP$,CNY$,TRX1$,TRX2$
PARAMETER (IJRI=100)
DIMENSION EPSCP$(IJRI),RS$(IJRI),EPSCF$(IJRI),
&           EPSXP$(IJRI),CNY$(IJRI),TRX1$(IJRI),TRX2$(IJRI)
DATA XCCP,XCCF,XCRP,XCRF,XALPHA,OTH,FTH,AA,BB,IXCH1
*           /10*1.D0/
DATA EPSCP,RS,EPSCF,EPSP,CNY,TRX1,TRX2
*           /7*1.D0/
DATA EPSCP$,RS$,EPSCF$,EPSP$,CNY$,TRX1$,TRX2$
*           /700*1.D0/
END

```

Fig. 4.12(a) (continued)

```

SUBROUTINE INIT9(NT,SPLIT,NEL,JRIS,GAR,RI,SGX,BGX,COR,GARS,
  NTMAX,JNTMAX)
  IMPLICIT REAL*8 (A-H,O-Z)
C->  DECLARE,DEFINE DATA
  DIMENSION SPLIT(NTMAX),NEL(NTMAX),JRIS(NTMAX),
  GAR(NTMAX,JNTMAX),RI(NTMAX,JNTMAX),
  SGX(NTMAX,JNTMAX),BGX(JNTMAX,NTMAX,NTMAX),
  COR(NTMAX,JNTMAX),GARS(JNTMAX,NTMAX,NTMAX)
  JRIS(1) = 1
  JRIS(2) = 100
  DO 3000 I=1,NT
    SPLIT(I) = 1.D0
    NEL(I) = 1
    DO 3100 J=1,JNTMAX,2
      COR(I,J)=1.D0
      GAR(I,J)=1.D0
      SGX(I,J)=0.D0
      RI(I,J)=1.D0
  3100 CONTINUE
    DO 3200 J=2,JNTMAX,2
      COR(I,J)=0.D0
      GAR(I,J)=1.D0
      SGX(I,J)=0.D0
      RI(I,J)=1.D0
  3200 CONTINUE
  3000 CONTINUE
    DO 3300 I=1,JNTMAX
    DO 3300 J=1,NT
    DO 3300 K=1,NT
      GARS(I,J,K)=1.0D0
      BGX (I,J,K)=1.0D0
  3300 CONTINUE
  RETURN
END

```

Fig. 4.12(a) (continued)

```

SUBROUTINE SUB12V(NT,SPLIT,NEL,JRIS,GAR,RI,SGX,BGX,COR,GARS
  ,NTMAX,JNTMAX,FAC1,FAC2)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON/CSUB12/ XCCP,XCCF,XCRP,XCRF,XALPHA,OTH,FTH,AA,BB,IXCH1
COMMON/DUMMY/EPSCP,RS,EPSCF,EPSXP,CNY,TRX1,TRX2
DIMENSION SPLIT(NTMAX),NEL(NTMAX),JRIS(NTMAX),
           GAR(NTMAX,JNTMAX),RI(NTMAX,JNTMAX),
           SGX(NTMAX,JNTMAX),BGX(JNTMAX,NTMAX,NTMAX),
           COR(NTMAX,JNTMAX),GARS(JNTMAX,NTMAX,NTMAX)
DIMENSION FAC1(NTMAX),FAC2(NTMAX)
RSF    = RS/XCRF
RSF2   = RSF*RSF
RSF3   = RSF2*RSF
RSP    = RS/XCRP
RSP2   = RSP*RSP
RSP3   = RSP2*RSP
FCF    = ( 1.00 + RSF3 ) * DLOG( 1.00 + 1.00/RSF )
*      + 0.500*RSF - RSF2 - OTH
FCP    = ( 1.00 + RSP3 ) * DLOG( 1.00 + 1.00/RSP )
*      + 0.500*RSP - RSP2 - OTH
EPSCP  = - XCCP*FCP
EPSCF  = - XCCF*FCF
EPSXP  = - .91633059D0/RS
CNY    = 5.1297628D0*( EPSCF - EPSCP )
ARS    = -1.22177412D0/RS + CNY
BRS    = - XCCP*DLOG( 1.00 + XCRP/RS ) - CNY
C
*VOCL LOOP,SCALAR
DO 24 IT=1,NT
  FAC1(IT)=SPLIT(IT)/NEL(IT)/2.00
  FAC2(IT)=0.500-FAC1(IT)
  JRI=JRIS(IT)
  IF(GAR(IT,JRI).LT.1.0-10) GAR(IT,JRI)=GAR(IT,JRI-1)*1.500+
  &                               GAR(IT,JRI-1)*0.500
24 CONTINUE
DO 26 IT=1,NT
  DO 25 IR=1,JRIS(IT)
    R=RI(IT,IR)
    RCE=R*R
    RHO=GAR(IT,IR)/RCE
    RHO1=0.500*RHO
    X     = RHO1/RHO
    FX    = ( X**FTH + ( 1.00 - X )**FTH - AA )/BB
    TRX1 = ( 2.00*X )**OTH
    CORX = ARS*TRX1 + BRS
C
C   GARS=4*PI*RADIUS**2*CHARGE DENSITY
C   BGX=RADIUS**2*POTENTIAL
C
    SGX(IT,IR)=RCE*(SGX(IT,IR)+CORX)
    BGX(IR,IT,1)=SGX(IT,IR)
    BGX(IR,IT,2)=SGX(IT,IR)
    GARS(IR,IT,1)=FAC1(IT)*GAR(IT,IR)+FAC2(IT)*GAR(IT,IR)
25 GARS(IR,IT,2)=COR(IT,IR)-GARS(IR,IT,1)
26 CONTINUE
RETURN
END

```

Fig. 4.12 (b) Vectorized program of problem no.12 by expert

```

PROGRAM EX13
IMPLICIT REAL*8 (A-H,O-Z)
C-> DECLARE,DEFINE DATA
PARAMETER (LCMAX=10,LCMAX2=100)
PARAMETER (IRMAX2=100)
REAL*8 A(LCMAX2*IRMAX2),B(LCMAX2*IRMAX2),C(LCMAX2*IRMAX2)
DATA LC,LC2,IR2/10,2*100/
C-----
DATA IOVER/1000/
OVERH = 0.D0
DO 1 I=1,IOVER
CALL CLOCK(STIME,2,2)
OVERH = OVERH + STIME
1 CONTINUE
OVERH = OVERH/IOVER
CALL CLOCK(STIME,2,2)
C-----
C-> ORIGINAL VERSION
DO 2 I=1,10000
CALL SUB13(LC,LC2,IR2,A,B,C)
2 CONTINUE
C-----
CALL CLOCK(TTIME,2,2)
S = (TTIME-STIME-OVERH)/10**6
WRITE(6,1000) S
1000 FORMAT(1H ,SUB13 TIME =',1PD15.3)
STOP
END
SUBROUTINE SUB13(LC,LC2,IR2,A,B,C)
REAL*8 A(1),B(1),C(1)
DO 420 K=1,LC
DO 410 KK=1,LC
DO 400 I=1,IR2
KX = K+ (KK-1)*LC+(I-1)*LC2
A(KX) = 0.D0
B(KX) = 0.D0
IF(K.EQ.KK) B(KX) = 1.D0
C(KX) = 0.D0
400 CONTINUE
410 CONTINUE
420 CONTINUE
RETURN
END

```

Fig. 4.13 (a) Original source program of problem no.13

```

SUBROUTINE SUB13V(LC,LC2,IR2,A,B,C)
REAL*8 A(1),B(1),C(1)
*VOCL LOOP,REPEAT(10)
DO 400 K=1,LC
*VOCL LOOP,REPEAT(10)
DO 400 KK=1,LC
*VOCL LOOP,REPEAT(100)
DO 400 I=1,IR2
KX = K+ (KK-1)*LC + (I-1)*LC2
A(KX) = 0.D0
B(KX) = 0.D0
C(KX) = 0.D0
400 CONTINUE
*VOCL LOOP,REPEAT(10)
DO 410 K=1,LC
*VOCL LOOP,REPEAT(100)
DO 410 I=1,IR2
KX = K+ (K-1)*LC + (I-1)*LC2
B(KX) = 1.D0
410 CONTINUE
RETURN
END

```

Fig. 4.13 (b) Vectorized program of problem no.13 by expert

```

PROGRAM EX14
IMPLICIT REAL*8 (A-H,O-Z)
C-> DECLARE,DEFINE DATA
PARAMETER ( L=5,IRMAXM =10)
DIMENSION LLMIN(L),LLMAX(L),A(300),B(300),C(300),
          RIM(IRMAXM,10),D2R(3,IRMAXM),ETA(IRMAXM,5)
DATA IRM/10/
DATA LCMAX,LCMAX2/5,25/
DATA LLMIN/1,3,2,4,5/
DATA LLMAX/3,5,5,4,5/
DATA LD,LCNT/1,3/
DATA BIMP,DTE,DTH/3*1.D0/
C-----
      DATA IOVER/1000/
      OVERH = 0.D0
      DO 1 I=1,IOVER
      CALL CLOCK(STIME,2,2)
      OVERH = OVERH + STIME
1 CONTINUE
      OVERH = OVERH/IOVER
      CALL CLOCK(STIME,2,2)
C-----
C-> ORIGINAL VERSION
DO 2 I=1,50000
      CALL SUB14(LLMIN,LLMAX,L,ETA,A,B,C,RIM,LD,IRMAXM,IRM,
                 LCNT,LCMAX,LCMAX2,BIMP,DTE,D2R,DTH)
2 CONTINUE
C-----
      CALL CLOCK(TTIME,2,2)
      S = (TTIME-STIME-OVERH)/10**6
      WRITE(6,1000) S
1000 FORMAT(1H , 'SUB14 TIME =',1PD15.3)
      STOP
      END
      SUBROUTINE SUB14(LLMIN,LLMAX,L,ETA,A,B,C,RIM,LD,IRMAXM,IRM,
                         LCNT,LCMAX,LCMAX2,BIMP,DTE,D2R,DTH)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION LLMIN(1),LLMAX(1),A(1),B(1),C(1),
                  RIM(IRMAXM,1),D2R(3,IRMAXM),ETA(IRMAXM,1)
      DO 145 K=LLMIN(L),LLMAX(L)
      KK = L-K+LCNT
      KKD=1+IABS(KK-LCNT)
      KX=LD+(KKD-1)*LCMAX
      DO 150 I=2,IRM
      KXX = KX+(I-2)*LCMAX2
      DTE = BIMP*DTH*ETA(I,K)
      A(KXX) = A(KXX)-DTE*D2R(1,I)
      B(KXX) = B(KXX)-DTE*(D2R(2,I)-RIM(I,K)**2)
      C(KXX) = C(KXX)-DTE*D2R(3,I)
150 CONTINUE
145 CONTINUE
      RETURN
      END

```

Fig. 4.14 (a) Original source program of problem no.14

```
SUBROUTINE SUB14V(LLMIN,LLMAX,L,ETA,A,B,C,RIM,LD,IRMAXM,IRM,
    .          LCNT,LCMAX,LCMAX2,BIMP,DTE,D2R,DTH)
    IMPLICIT REAL*8 (A-H,O-Z)
    DIMENSION LLMIN(1),LLMAX(1),A(1),B(1),C(1),
    .          RIM(IRMAXM,1),D2R(3,1),ETA(IRMAXM,1)
    DO 145 K=LLMIN(L),LLMAX(L)
*VOCL LOOP,REPEAT(11)
*VOCL LOOP,LCMAX2.GT.0
    DO 150 I=2,IRM
        KXX = LD+IABS(L-K)*LCMAX+(I-2)*LCMAX2
        DTE = BIMP*DTH*ETA(I,K)
        A(KXX) = A(KXX)-DTE*D2R(1,I)
        B(KXX) = B(KXX)-DTE*(D2R(2,I)-RIM(I,L-K+LCNT)**2)
        C(KXX) = C(KXX)-DTE*D2R(3,I)
150 CONTINUE
145 CONTINUE
    RETURN
END
```

Fig. 4.14 (b) Vectorized program of problem no.14 by expert

5. おわりに

これまで50本以上の原子力コードのベクトル化で蓄積されてきたベクトル化ノウハウを解析し体系化するルールベース・システムのプロトタイプを作成した。これによって、ベクトル化経験の少ない者が簡単なプログラムに対して、コンパイラの自動ベクトル化機能ではベクトル化できない理由やベクトル化はされてもさらに効率よく処理するためのチューニングのための対策を推奨することができるようになった。またCes-VPの試用調査を行い性能とマン・マシン・インターフェースの問題点や今後の課題を検討し、今後このシステムを実用システムとするためには、

- プログラム解析ツールやベクトル化効果表示ツールとの結合
- 推奨するベクトル化・チューニング技法についての具体例を表示
- ルールの学習機能

を検討していく事が重要であることがわかった。さらに、Ces-VPを原子力コードのベクトル化支援のため原子力コード特有の知識を持つシステムへの発展を考えることも重要である。現在のCes-VPは、FORTRAN 77／VPの自動ベクトル化機能を補足するようなベクトル化の知識やベクトル化処理効率の向上の知識がルールとして表現されているが、例えば、原子炉を構成する部品の数を知識として持ち、あるDOループが、どの部品に関する計算をするのかを質問することによって、そのDOループの繰り返し回数を予測しへクトル処理をするかどうかを判断するという原子力コードに関する知識を利用してプログラムの意味を考えたベクトル化の支援をするエキスパート・システムを考えることも重要である。

謝 辞

Ces-VPの開発に先立ちベクトル化メッセージによるベクトル化診断エキスパート・システムについて有意義な議論をしてくださった富士通株式会社石谷 隆広氏に感謝します。

アンケート調査のご協力頂きました計算センターの五来 一夫氏、外来研究員の野々宮 厳氏、根本 俊行氏、染谷 千浩氏、財団法人原子力データセンタの中野 美貴志氏、千葉 猛美氏、山崎 隆氏、前村 克己氏、雨貝 彰氏、鶴岡 卓哉氏、堀 肇雄氏、中村 光行氏に感謝します。

本報告書を書く機会を与えて頂きました計算センター室長浅井 清氏及び富士通株式会社科学システム部第2システム課長南 多善氏に感謝します。

5. おわりに

これまで50本以上の原子力コードのベクトル化で蓄積されてきたベクトル化ノウハウを解析し体系化するルールベース・システムのプロトタイプを作成した。これによって、ベクトル化経験の少ない者が簡単なプログラムに対して、コンパイラの自動ベクトル化機能ではベクトル化できない理由やベクトル化はされてもさらに効率よく処理するためのチューニングのための対策を推奨することができるようになった。またCes-VPの試用調査を行い性能とマン・マシン・インターフェースの問題点や今後の課題を検討し、今後このシステムを実用システムとするためには、

- プログラム解析ツールやベクトル化効果表示ツールとの結合
- 推奨するベクトル化・チューニング技法についての具体例を表示
- ルールの学習機能

を検討していく事が重要であることがわかった。さらに、Ces-VPを原子力コードのベクトル化支援のため原子力コード特有の知識を持つシステムへの発展を考えることも重要である。現在のCes-VPは、FORTRAN 77／VPの自動ベクトル化機能を補足するようなベクトル化の知識やベクトル化処理効率の向上の知識がルールとして表現されているが、例えば、原子炉を構成する部品の数を知識として持ち、あるDOループが、どの部品に関する計算をするのかを質問することによって、そのDOループの繰り返し回数を予測しへクトル処理をするかどうかを判断するという原子力コードに関する知識を利用してプログラムの意味を考えたベクトル化の支援をするエキスパート・システムを考えることも重要である。

謝 辞

Ces-VPの開発に先立ちベクトル化メッセージによるベクトル化診断エキスパート・システムについて有意義な議論をしてくださった富士通株式会社石谷 隆広氏に感謝します。

アンケート調査のご協力頂きました計算センターの五来 一夫氏、外来研究員の野々宮 厳氏、根本 俊行氏、染谷 千浩氏、財団法人原子力データセンタの中野 美貴志氏、千葉 猛美氏、山崎 隆氏、前村 克己氏、雨貝 彰氏、鶴岡 卓哉氏、堀 肇雄氏、中村 光行氏に感謝します。

本報告書を書く機会を与えて頂きました計算センター室長浅井 清氏及び富士通株式会社科学システム部第2システム課長南 多善氏に感謝します。

参考文献

- 1) 石黒美佐子, 松浦俊彦, 奥田 基, 原田裕夫, 太田文男, 梅谷 真; ベクトル計算処理の大型原子力コードへの適応性, JAERI-M 82-018, 1982.
- 2) 原田裕夫, 橋口健二, 石黒美佐子, 筒井恒夫, 藤井 実; FACOM 230-75 APUによる原子力コードのベクトル化, JAERI-M 83-024, 1983.
- 3) 浅井 清; スーパーコンピュータの動向と原子力コード分野におけるベクトル計算処理, 第16回「炉物理夏の学校」テキスト, P67, 1984年7月24日~27日.
- 4) 石黒美佐子; 原子力コードのベクトル化経験, 原子力コードにおけるソフトウェア開発研究会報告書, JAERI-M 85-017, 1985.
- 5) 奈良岡賢逸, 德永康男, 栗田 豊, 能村博人, 篠沢尚久; 原子力コードのベクトル化プログラミング [I], 私信.
- 6) 德永康男, 原田裕夫, 石黒美佐子; 原子力コードにおける数値解法とそのベクトル化, JAERI-M 85-143, 1985.
- 7) M. Ishiguro, H. Harada and M. Makino; Performance Analysis of vectorized Nuclear Codes on a FACOM VP-100 at the Japan Atomic Energy Research Institute, The International Journal of Supercomputer Applications, Vol.1 No.3, 1987, pp44-
- 8) 日本原子力研究所計算センタ; 原子力コードのベクトル化基礎 (FACOM FORTRAN77／VP V10 L20 対応) 私信.
- 9) 富士通(株); FACOM OS IV/F4 MSP FORTRAN77／VP プログラミング・ハンドブック, 1985.
- 10) 富士通(株); FACOM EHELL 解説書, 1986.
- 11) 富士通(株); FACOM FORTUNE 使用手引書, 1987.
- 12) 藤崎正英, 牧野光弘, 石黒美佐子; 原子力コードのベクトル化支援ツール Ces-VP, 情報処理学会, 第35回全国大会, 1987.

付録 知識ベースのルール

KS name	KS-INIT (FORWARD)	診断内容選択KS
trigger	INIT	
strategy	MULTIPLE MANYTIMES	
RULE NO.	L (条) H (件) S (部)	R (実) H (行) S (部)
1	(EQUAL (@VALUE 'CONTROL-TOTAL 'FIRST) 'NO) 診断内容選択KSを通過したのは、初めてでない。	(1) EXECUTE EXPRESSION : (PROGN (CLEAR-NODE '@DATA) (CLEAR-NODE 'ND-TUNNING)) DBのSTATIC NODE @DATA, ND-TUNNING の値をNILでクリアする。 (2) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING ATTRIBUTES : CONCLUSION=0 ベクトル化・チューニング制御フラグを クリアする。
2	(EQUAL (@VALUE 'CONTROL-TOTAL 'FIRST) 'YES) 診断内容選択KSを通過したのは、初めてである。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'CONTROL-TOTAL ATTRIBUTE : FIRST ='NO 診断内容選択KSの通過フラグをNOにする
3	(EQUAL (@VALUE '@DATA 'QACONSULT) 1) 診断内容の選択番号は1である。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : VP NODE : 'CONTROL-TOTAL ベクトル化・チューニング制御KSを 発行する。

KS name	KS-VP (FORWARD)	ベクトル化・チューニング制御KS
trigger	VP	
strategy	MULTIPLE MANYTIMES	
RULE NO.	L H S (条 件 部)	R H S (実 行 部)
1	<p>(EQUAL (@VALUE 'CONTROL-VP 'FIRST) 'NO)</p> <p>ベクトル化・チューニング制御KSを通過したのは初めてでない。</p>	<p>(1) EXECUTE EXPRESSION : (PROGN (CLEAR-NODE '@DATA) (CLEAR-NODE 'ND-TUNNING)) BBのSTATIC NODE @DATA, ND-TUNNING の値をNILでクリアする。</p> <p>(2) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : IF-INTEGRATION = 'NO VOCL-IF = 'NO LIST-PROCESSING = 'NO ACCESS-CONFLICT = 'NO BANK-CONFLICT = 'NO WORK-DIMENSION = 'NO LOOP-TAKE-IN = 'NO PARTITION = 'NO INTEGRATION = 'NO VOCL-IF = 'NO MONO = 'NO MONO-VAR = 'NO WRITE-DOWN = 'NO INTERCHANGE = 'NO INTERCHANGE-VAR = NIL UNROLLING = 'NO UNROLLING-VAR = NIL DIVISION = 'NO VOCL-SCALAR = 'NO VOCL-REPEAT = 'NO DELTA = 'NO DELTA-VAR = NIL J227-COMMENT = NIL J227-INDEPENDENT = NIL J228-EXPAND = NIL J228-DIVIDE = NIL J229-CHARACTER = NIL J231-DIVIDE = 'NO J231-INTERCHANGE = 'NO J231-VOCL-NOVREC = 'NO J249-DIVIDE = 'NO J249-INTERCHANGE = 'NO J249-NOKS = 'NO J249-OTHER = 'DUMMY J250-DIVIDE = 'NO J250-INTERCHANGE = 'NO J262-INTERCHANGE = 'NO</p>

	<pre>J262-NOKS = 'NO J273-DIVIDE = 'NO J273-INTERCHANGE = 'NO J273-VOCL-NOVREC = 'NO J277-WRITE-DOWN = 'NO J277-WORK = 'NO</pre> <p>チューニングのBBをデフォルト値にクリアする。</p>
	<p>(3) PROPOSE</p> <pre>CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING ATTRIBUTES : CONCLUSION=0</pre> <p>ベクトル化・チューニング制御フラグをクリアする。</p>
2	<p>(EQUAL (@VALUE 'CONTROL-VP 'FIRST) 'YES)</p> <p>ベクトル化・チューニング制御KSを通過したのは初めてである。</p> <p>(1) EXECUTE</p> <pre>EXPRESSION : (PROGN (ERASE) (FORMAT " /N /C /N " " 使用方法はつきの通りです。) (FORMAT " /N /C /N " " 1.会話型ベクトライザのバッチ出力を用意してください。) (FORMAT " /N /C /N /C /N " " 2.一番コストの高いプログラム単位（サブルーチンなど）" " から、コスト順に以下の手順を繰り返してください。 ") (FORMAT " /N /C /N /C /N " " 3.一番コストの高いDOループからコスト順に、あるいは、" " 上から順に以下の手順を繰り返してください。 ") (FORMAT " /N /C /N /C /N " " 4. DOループに対するベクトル化メッセージ及びループ構造" " を調べて、質問に答えてください。 ") (FORMAT " /N /C /N /C /N " " 尚、質問には、特に指定の無い限り、YESまたは、NO" " で答えてください。 ") (WAIT " ENTER KEY を押してください。 ") (ERASE))</pre> <p>使用方法の説明をする。</p> <p>(2) PROPOSE</p> <pre>CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'CONTROL-VP ATTRIBUTES : FIRST ='NO</pre> <p>ベクトル化・チューニング制御KSの通過フラグをNOにする。</p>

3	<pre>(EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'YES)</pre> <p>ベクトル化表示は、Vになっている。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : TUNING NODE : 'ND-TUNNING ATTRIBUTES : CONCLUSION=2 ベクトル化・チューニング制御フラグを2にする。(チューニングのみ)</p> <p>(2) EXECUTE EXPRESSION : (ERASE) 画面を消去する。</p>
4	<pre>(NOT (EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'QUIT)) (NOT (EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'END)) (EQUAL (@VALUE '@DATA 'QAJND) 'YES) (@VALUE '@DATA 'QAJNDNNN)</pre> <p>ベクトル化メッセージが一つ以上出力され、診断を終了させない。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : VECTORIZE NODE : 'ND-TUNNING ATTRIBUTES : CONCLUSION=1 ベクトル化・チューニング制御フラグを1にする。(ベクトル化のみ)</p> <p>(2) EXECUTE EXPRESSION : (ERASE) 画面を消去する。</p>
5	<pre>(EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'YES) (EQUAL (@VALUE '@DATA 'QAJND) 'YES) (@VALUE '@DATA 'QAJNDNNN)</pre> <p>ベクトル化メッセージが一つ以上出力され、かつベクトル化表示は、Vになっている。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : VECTORIZE NODE : 'ND-TUNNING ATTRIBUTES : CONCLUSION=3 ベクトル化・チューニング制御フラグを3にする。(チューニング・ベクトル化両方)</p>
6	<pre>(OR (AND (EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'NO) (EQUAL (@VALUE '@DATA 'QAJND) 'NO)) (AND (NOT (EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'QUIT)) (NOT (EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'END)) (<= (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) 0))))</pre> <p>ベクトル化メッセージが出力されず、ベクトル化表示がVになっていない。または、診断を終了しないがDOループがない。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : INIT NODE : 'CONTROL-TOTAL 診断内容選択KSを発行する。</p> <p>(2) EXECUTE EXPRESSION : (PROGN (NOKS) (WAIT "ENTER KEY を押してください")))</p> <p>現在のKSでは、診断できない旨を出力する</p>
7	<pre>(EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'QUIT)</pre> <p>診断の初期メニューにもどる。</p>	<p>(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : INIT NODE : 'CONTROL-TOTAL 診断内容選択KSを発行する。</p>

<pre> 8 (NOT (EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'QUIT)) (NOT (EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'END)) (NOT (AND (EQUAL (@VALUE '@DATA 'QAVECTORIZED) 'NO) (EQUAL (@VALUE '@DATA 'QAJND) 'NO))) (> (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) 0) </pre> <p> ① 診断を終了しない。 ② ベクトル化表示がVになっている。または、 ベクトル化メッセージが出力されている。 ③ DO ループが存在する。 ①～③を同時に満たす。 </p>	<pre> (I) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING ATTRIBUTES : LOOP-LEN = (ASK-DOLIST NIL (@VALUE 'ND-TUNNING 'QA-MULTIPLEX)) VECTOR-LEN = (COND ((EQUAL (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) 1) (NTH 0 (REVERSE (@VALUE 'ND-TUNNING 'LOOP-LEN)))) (T (NTH (SUB1 (@VALUE '@DATA 'QA-VLOOP) (REVERSE (@VALUE 'ND-TUNNING 'LOOP-LEN))))) CHG-VAR = (CHG-DOVAR (@VALUE 'ND-TUNNING 'LOOP-LEN) (@VALUE 'ND-TUNNING 'VECTOR-LEN)) ALL-VAR = (ALL-DOVAR (@VALUE 'ND-TUNNING 'LOOP-LEN) (@VALUE 'ND-TUNNING 'VECTOR-LEN)) DIMENSION-SOEJI = (ASK-SOEJI (@VALUE 'ND-TUNNING 'LOOP-LEN) (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) (@VALUE '@DATA 'QA-DIM-NUMBER)) </pre> <p>DOループの繰り返し回数、ベクトル化ループの指定、DO ループの入れ替え候補の組合せ、全てのDOループ変数の組合せをセットする。</p>
--	--

KS name	KS-TUNING (FORWARD)	チューニング診断KS
trigger	TUNING	
strategy	MULTIPLE MANYTIMES	
LOCAL-VARIABLE	EXPRESSION	
L-IFSTMT L-MULT	(@VALUE '@DATA 'QAIFSTMT) (@VALUE 'ND-TUNNING 'QA-MULTIPLEX)	
RULE NO.	L H S (条 件 部)	R H S (実 行 部)
1	(EQUAL L-IFSTMT 'YES) (>= (@VALUE '@DATA 'QAIFNEST) 4) 入れ子が4重以上のIF文がある。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : IF-INTEGRATION = 'YES IF文の統合フラグをYESにする。
2	(EQUAL L-IFSTMT 'YES) (EQUAL (@VALUE '@DATA 'QAIFTRUTH) 'YES) IF文の真率が既知である。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : VOCL-IF = 'YES VOCL IF文の挿入フラグをYESにする。
3	(EQUAL (@VALUE '@DATA 'QAIFSTMT) 'YES) (EQUAL (@VALUE '@DATA 'QAIFGOTO-OUT) 'YES) DOループの外に飛び出すIF～GOTO文が二つ以上ある。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : LIST-PROCESSING = 'OUT リストベクトルフラグをOUTにする。
4	(EQUAL (@VALUE '@DATA 'QA-UPGOTO) 'YES) 上向きのGOTO文がある。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : LIST-PROCESSING = 'UPGOTO リストベクトルフラグをUPGOTOにする
5	(EQUAL L-IFSTMT 'YES) (>= (@VALUE '@DATA 'QAIFNEST) 4) 入れ子が4重以上のIF文がある。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : LIST-PROCESSING = 'NEST リストベクトルフラグをNESTにする。

6	<p>(EQUAL (@VALUE '@DATA 'QA-MEM) 'YES)</p> <p>配列の添字は、配列要素であり同じ値をとすることが多い。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : ACCESS-CONFLICT = 'YES メモリーのアクセス・コンフリクトのフラグをYESにする。</p>
7	<p>(OR (AND (EQUAL (@VALUE '@DATA 'QA-DIM-TYPE) 'R*4) (NOT (= (REMAINDER (@VALUE '@DATA 'QA-DIM-SOEJI) 4) 2))) (AND (EQUAL (@VALUE '@DATA 'QA-DIM-TYPE) 'R*8) (NOT (= (REMAINDER (@VALUE '@DATA 'QA-DIM-SOEJI) 2) 1)))))</p> <p>実数型4バイトの配列の第一添字が4N+2以外で宣言されているか、実数型8バイトの配列の第一添字が2N+1以外で宣言されている。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : BANK-CONFLICT = 'YES バンク・コンフリクトフラグをYESにする。</p>
8	<p>(EQUAL (@VALUE '@DATA 'QA-WORK-DIM) 'YES)</p> <p>DOループの外で引用されてない中間配列がある。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : WORK-DIMENSION = 'YES スカラー変数フラグをYESにする。</p>
9	<p>(EQUAL (@VALUE '@DATA 'QA-UPDO) 'YES)</p> <p>対象のルーチンを呼び出すCALL文は、全て同じ繰り返し回数のDOループ内にあり、かつそのCALL文の前後でDOループが分割できる。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : LOOP-LEN = (ASK-DOLIST (@VALUE 'ND-TUNNING 'LOOP-LEN)) (@VALUE '@DATA 'QA-UP-DONUM)) UP-LOOP = (@VALUE 'ND-TUNNING 'LOOP-LEN)) QA-MULTIPLEX = (PLUS (@VALUE 'ND-TUNNING 'QA-MULTIPLEX)) (@VALUE '@DATA 'QA-UPDONUM)) CHG-VAR = (APPEND (CHG-DOVAR (NTHCAR (@VALUE '@DATA 'QA-UP-DONUM)) (@VALUE 'ND-TUNNING 'LOOP-LEN)) (@VALUE 'ND-TUNNING 'VECTOR-LEN)) (@VALUE 'ND-TUNNING 'CHG-VAR)) ALL-VAR = (APPEND (ALL-DOVAR (NTHCAR (@VALUE '@DATA 'QA-UP-DONUM)) (@VALUE 'ND-TUNNING 'LOOP-LEN)) (@VALUE 'ND-TUNNING 'VECTOR-LEN)) (@VALUE 'ND-TUNNING 'ALL-VAR))</p>

		DOループの持ち込みによるDOループの特徴を表す属性値の変更
		(2) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : LOOP-TAKE-IN='YES DOループの持ち込みフラグをYESにする。
10	(EQUAL (@VALUE '@DATA 'QAVSM) 'YES) (< (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) 2) ベクトル化表示は、V, S, Mが混在している。 DOループの多重度が1である。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : PARTITION = 'VSM DOループ分割フラグをVSMにする。
11	(EQUAL (@VALUE '@DATA 'QAVSM) 'YES) (>= (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) 2) ベクトル化表示は、V, S, Mが混在している。 DOループの多重度が2以上である。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : PARTITION = 'VSM DOループ分割フラグをVSMにする。
12	(>= (@VALUE '@DATA 'QA-PART #STMT) 100) DOループ内のステートメント数が100以上である。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : PARTITION = 'STMT DOループ分割フラグをSTMTにする。
13	(EQUAL (@VALUE '@DATA 'QA-IFGOTO-OUT) 'YES) DOループの外に飛び出すIF～GOTO文が二つ以上ある。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : PARTITION = 'OUT DOループ分割フラグをOUTにする。
14	(EQUAL (@VALUE '@DATA 'QA-INTEGRATE#STR) 'YES) (EQUAL (@VALUE '@DATA 'QA-INTEGRATE#IFGOTO) 'NO) (EQUAL (@VALUE '@DATA 'QA-INTEGRATE#REC) 'NO) 論理的に隣合ったDOループにおいて、それぞれのDO文の構成が同じで、DOループの統合によって、IF～GOTO文が二つ以上にならず参照関係も変わらない。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : INTEGATION = 'YES DOループ統合フラグをYESにする。
15	(> (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) 1) (<= (NTH 2 (CAR (LAST (EQUAL (@VALUE 'ND-TUNNING 'LOOP-LEN)))) 7)) (@VALUE 'ND-TUNNING 'CHG-VAR) (OR (EQUAL (NTH 3 (CAR (LAST (@VALUE 'ND-TUNNING 'LOOP-LEN)))) 1) (EQUAL (NTH 3 (CAR (LAST (@VALUE 'ND-TUNNING 'LOOP-LEN)))) 1.0)) 多重DOループで、最内DOループの繰り返し回数が7より小さくて、その確信度が1（繰り返し回数が定数）である。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : WRITE-DOWN = 'YES 最内DOループの書き下しフラグをYESにする。

16	<pre> ITERATION : L-CHG = (@VALUE 'ND-TUNNING 'CHG-VAR) (> (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) 1) (EQUAL (NTH 2 L-CHG) 'NO) (BEQUAL (@VALUE '@DATA 'QAGOTO-OUT) 'NO) (EQUAL (ASK-CHG (NTH 0 L-CHG) (NTH 1 L-CHG)) 'NO) (OR (AND (>= (LENGTH (VL-EQ (@VALUE 'ND-TUNNING 'LOOP-LEN) (@VALUE 'ND-TUNNING 'VECTOR-LEN)))) 1) (= (ACC-GOOD (@VALUE 'ND-TUNNING 'DIMENSION-SOBJI) (VL-EQ (@VALUE 'ND-TUNNING 'LOOP-LEN) (@VALUE 'ND-TUNNING 'VECTOR-LEN))) (LIST (NTH 1 (@VALUE 'ND-TUNNING 'VECTOR-LEN)))) 1) (>= (LENGTH (VL-GT (@VALUE 'ND-TUNNING 'LOOP-LEN) (@VALUE 'ND-TUNNING 'VECTOR-LEN)))) 1)) </pre> <p>多重DOループで入れ替えによって、参照関係に矛盾がおきず、メモリーアクセスが良くなつてかつベクトル長が同じか、または、メモリーアクセスが悪くならずかつベクトル長が大きくなる。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : INTERCHANGE = 'YES INTERCHANGE-VAR = L-CHG</p> <p>DOループの入れ替えフラグをYESにして、そのDOループの変数の組をセットする。</p>
17	<pre> ITERATION : L-MONO = (@VALUE 'ND-TUNNING 'CHG-VAR) (> (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) 1) (EQUAL (NTH 2 L-MONO) 'NO) (EQUAL (ASK-MONO (NTH 0 L-MONO) (NTH 1 L-MONO)) 'YES) </pre> <p>多重DOループで入れ替えによって、参照関係に矛盾がおきず、DOループの外に飛び出すGOTO文がなく各配列の添字式式中のみに同じ形式・順序で出現していて、かつDOループ実行後に引用されていない。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : INTERCHANGE = 'YES INTERCHANGE-VAR = L-CHG</p> <p>DOループの一重化のフラグをYESにして、そのDOループの変数の組をセットする。</p>
18	<pre> ITERATION : L-UNR = (@VALUE 'ND-TUNNING 'ALL-VAR) (> (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) 1) (TONARI (@VALUE 'ND-TUNNING 'LOOP-LEN) (NTH 0 L-UNR) (NTH 1 L-UNR)) </pre> <p>多重DOループでベクトル化されていないDOループが存在する。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : UNROLLING = 'YES UNROLLING-VAR = L-UNR</p> <p>アンローリングのフラグをYESにして、そのDOループの変数の組をセットする。</p>
19	<pre> (EQUAL (@VALUE '@DATA 'QALPMAX) 'YES) </pre> <p>DOループの繰り返し回数の最大値が既知である。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : VOCL-REPEAT = 'YES VOCL REPEAT文挿入フラグをYESにする。</p>

20	(EQUAL (@VALUE '@DATA 'QA-DIVIDE) 'YES) ベクトル長が可変で、かつ小さいときが多い。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : DIVISION = 'YES スカラーループとベクトルループの分割フラグをYESにする。
21	(OR (AND (< (NTH 2 (@VALUE 'ND-TUNNING 'VECTOR-LEN)) 7) (EQUAL L-IFSTMT 'YES)) (AND (< (NTH 2 (@VALUE 'ND-TUNNING 'VECTOR-LEN)) 5) (EQUAL L-IFSTMT 'NO)) (AND (> (@VALUE '@DATA 'QA-PART #STMT) 100) (EQUAL (@VALUE 'ND-TUNNING-CONTROL 'PARTITION) 'NO))) ベクトル長は7より小さくIF文がある。または、 ベクトル長は5より小さくIF文がない。または、 DOループ内のステートメント数は100以上である のに、DOループの分割ができない。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : VOCAL-SCALAR = 'YES VOCAL SCALAR文挿入フラグをYESにする。
22	ITERATION : L-DELTA = (@VALUE 'ND-TUNNING 'LOOP-LEN) (NOT (EQUAL (NTH 4 L-DELTA) 1)) DOループのDO変数の増分値が1以外である。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : DELTA = 'YES DELTA-VAR = (CONS (NTH 1 L-DELTA) (@VALUE 'ND-TUNNING-CONTROL 'DELTA-VAR)) DO変数の増分値修正フラグをYESにして、 そのDOループの変数をセットする。
23	(NOT (AND (OR (EQUAL (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (EQUAL (@VALUE 'ND-TUNNING 'CONCLUSION) 3) (MEMBER 273 (@VALUE '@DATA 'QAJNDNNN))) ベクトル化・チューニング制御フラグが1でも3でもないか、または、ベクトル化メッセージに273が含まれていない。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : TUNING-BND NODE : 'ND-TUNNING-CONTROL チューニング結果出力KSを発行する。

2	(OR (EQUAL (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (EQUAL (@VALUE 'ND-TUNNING 'CONCLUSION) 3))	(1) EXECUTE EXPRESSION : (ERASE) 画面を消去する。
	ベクトル化・チューニング制御フラグが1かまたは3である。	(1) EXECUTE EXPRESSION : (PROGN (FG : DISPLAY (" * * * * * * * * * * * * * * * * * " 2 3 HB) (" * * ベクトル化診断結果 * * " 3 3 HB) (" * * * * * * * * * * * * * * * * * " 4 3 HB)) (FORMAT " /N つぎに示すような点について検討ください。/N ") (TERPRI TERMINAL-OUTPUT))
3	(EQUAL (@VALUE '@DATA 'QAJND) 'YES) (MEMBER 273 (@VALUE '@DATA 'QAJNDNNN)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1))	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /N /C /N /N /C /N /C " " < メッセージ番号JND-273 の意味 >" " オーバーヘッド(ベクトル, スカラー間の転送など)が大きすぎる." " ** メッセージ番号JND-273 に対する処置 **" " チューニング診断をおこなうか, 他のメッセージを先に解決してください。") (TERPRI TERMINAL-OUTPUT))
	(= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))	ベクトル化メッセージが273が出力されている。
4	(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J273-D IVIDE) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1))	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " 回帰参照でベクトル化できない部分と その他のベクトル化できる部分にDOループを分割してください。") (TERPRI TERMINAL-OUTPUT))
	(= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))	ベクトル化メッセージ273に関する ベクトルループとスカラーループの分割フラグがYESである。
5	(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J273-I INTERCHANGE) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1))	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " DOループの入れ替えによる回帰参照の 除去を検討してください。") (TERPRI TERMINAL-OUTPUT))
	(= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))	ベクトル化メッセージ273に関する DOループの入れ替えフラグがYESになっている。

6	(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J273-V DCL-NOVREC) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ273 に関連する VOCL NOVREC 文の挿入フラグがYES になっている。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " DO文の直前に * VOCL NOVREC 文を挿入 して下さい。") (TERPRI TERMINAL-OUTPUT))
7	(EQUAL L-IF-INTEGRATION) 'YES) IF文の統合フラグがYES である。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C " " IF文を統合して IF文の入れ子を3重以 下になるように検討して下さい。") (TERPRI TERMINAL-OUTPUT))
8	(EQUAL L-VOCL-IF 'YES) VOCL IF 文の挿入フラグがYES である。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C " " IF文の直前に * VOCL STMT, IF (真率) を挿入してください。") (TERPRI TERMINAL-OUTPUT))
9	(EQUAL L-LIST-PROCESSING 'OUT) リストベクトルフラグがOUT である。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " DOループの外に飛び出す配列要素のリ ストを作成し, " " 外に飛び出すGOTO文を なくすように検討してください。") (TERPRI TERMINAL-OUTPUT))
10	(EQUAL L-LIST-PROCESSING 'UPGOTO) リストベクトルフラグがUPGOTOである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " 上向きのGOTO文に対応する配列要素のリ ストを作成し, " " 上向きのGOTO文をな くすように検討してください。") (TERPRI TERMINAL-OUTPUT))
11	(EQUAL L-LIST-PROCESSING 'NEST) リストベクトルフラグがNESTである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " 多重のIF文の入れ子に対応する配列要 素のリストを作成し, " " 多重度を減らす ように検討してください。") (TERPRI TERMINAL-OUTPUT))

12	(EQUAL L-ACCESS-CONFLICT 'YES) アクセス・コンフリクトフラグがYESである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " 同じ値をとすることが多い配列の添字は、 あらかじめ作業配列に蓄えて”おき、 その作業配列を使ってアクセスするよう に検討してください。”) (TERPRI TERMINAL-OUTPUT))
13	(EQUAL L-BANK-CONFLICT 'YES) (@VALUE '@DATA 'QA-DIM-TYPE) 'R*4) バンク・コンフリクトフラグがYESであり、かつ配列が実数型4バイトで宣言されている。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " 実数型4バイトの配列の第1添字は、 4 N + 2 (Nは整数)で”宣言するよう に検討してください。”) (TERPRI TERMINAL-OUTPUT))
14	(EQUAL L-BANK-CONFLICT 'YES) (@VALUE '@DATA 'QA-DIM-TYPE) 'R*8) バンク・コンフリクトフラグがYESであり、かつ配列が実数型8バイトで宣言されている。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " 実数型8バイトの配列の第1添字は、 2 N + 1 (Nは整数)で”宣言するよう に検討してください。”) (TERPRI TERMINAL-OUTPUT))
15	(EQUAL L-WORK-DIMENSION 'YES) 作業配列フラグがYESである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " ベクトル化ループのみで使用される中 間配列は、スカラー変数に”修正するよ うに検討してください。”) (TERPRI TERMINAL-OUTPUT))
16	(EQUAL L-LOOP-TAKE-IN 'YES) DOループの持ち込みフラグがYESである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N 上位ルーチンの/C /C /N /C " " (MAPCAN (NTHCAR (@VALUE '@DATA 'QA-UP-DONUM) (@VALUE 'ND-TUNNING 'UP-LOOP)) (FUNCTION (LAMBDA (R) (LIST (NTH 1 R))))) ”に関するループを持ち込み、上位ルー チンでは、このサブルーチン”の前後で DOループを分割してください。”) (TERPRI TERMINAL-OUTPUT))
17	(EQUAL L-PARTITION 'VSM) DOループ分割フラグがVSMである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C " " ベクトル化部分と非ベクトル化部分の 分割を検討してください。”) (TERPRI TERMINAL-OUTPUT))

18	(EQUAL L-PARTITION 'STMT)	<p>DOループ分割フラグがSTMTである。</p> <p>(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " 1つのDOループの中のステートメント 数を50以下にするようなDOループ" の分 割を検討してください。") (TERPRI TERMINAL-OUTPUT))</p>
19	(EQUAL L-PARTITION 'OUT)	<p>DOループ分割フラグがOUTである。</p> <p>(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " DOループの外に飛び出すIF～GOTO文が 真になるリストをつくるループと" "その リストを使って処理をおこなうループに分 割できるかを検討してください。") (TERPRI TERMINAL-OUTPUT))</p>
20	(EQUAL L-INTEGRATION 'YES)	<p>DOループ統合フラグがYESである。</p> <p>(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C /N /C " " 論理的に隣合い且つ構成 (DOループの 繰り返し回数など) の同じDOループ" "の 統合を検討してください。") (TERPRI TERMINAL-OUTPUT))</p>
21	(EQUAL L-MONO 'YES)	<p>DOループ一重化フラグがYESである。</p> <p>(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C に関するDOループと /C に関 するDOループ /N /C " (CAR L-MONO-VAR) (CADR L-MONO-VAR) " の一重化を検討してください。") (TERPRI TERMINAL-OUTPUT))</p>
22	(EQUAL L-WRITE-DOWN 'YES)	<p>最内ループ書き下しフラグがYESである。</p> <p>(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C " " 最内DOループの書き下しを検討してく ださい。") (TERPRI TERMINAL-OUTPUT))</p>
23	(EQUAL L-INTERCHANGE 'YES)	<p>DOループ入れ替えフラグがYESである。</p> <p>(1) EXECUTE EXPRESSION : (PROGN (FORMAT " /N /C に関するDOループと /C に関 するDOループ /N /C " (CAR L-INTERCHANGE-VAR) (CADR L-INTERCHANGE-VAR) " の入れ替えを検討してください。") (TERPRI TERMINAL-OUTPUT))</p>

24	(EQUAL L-UNROLLING 'YES) アンローリングフラグがYESである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT "/N/C"に関するDOループについての アンローリングを検討してください。") (CAR L-UNROLLING-VAR)) (TERPRI TERMINAL-OUTPUT))
25	(EQUAL L-DIVISION 'YES) ベクトル化ループと非ベクトル化ループの分割 フラグがYESである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT "/N/C/N/C/N/C" "ベクトル長が7以下のときは、*VOCL LOOP, SCALAR文を" "DO文の直前に挿入した DOループを通り、7より大きいときはベクトル" "処理を行うような修正を検討してく ださい。") (TERPRI TERMINAL-OUTPUT))
26	(EQUAL L-PARTITION 'VSM2) 多重DOループの分割フラグがYESである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT "/N/C" "ベクトル化部分と非ベ クトル化部分の分割を検討して下さい。") (TERPRI TERMINAL-OUTPUT))
27	(EQUAL L-VOCL-SCALAR 'YES) VOCL SCALAR 文の挿入フラグがYESである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT "/N/C" "*VOCL LOOP, SCALAR文 をDO文の直前に挿入してください。") (TERPRI TERMINAL-OUTPUT))
28	(EQUAL L-VOCL-REPEAT 'YES) VOCL REPEAT 文の挿入フラグがYESである。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT "/N/C" "*VOCL LOOP, REPEAT (DO ループの繰り返し回数の最大値)文" " をDO文の直前に挿入してください。") (TERPRI TERMINAL-OUTPUT))
29	ITERATION: L-DVAR = L-DELTA-VAR (EQUAL L-DELTA 'YES) DO変数の増分値が1以外で指定してある。	(1) EXECUTE EXPRESSION : (PROGN (FORMAT "/N/C/C/C/N" L-DVAR "に関するDO文における増分値を正(なる べく1)にするような" "修正を検討して ください。") (TERPRI TERMINAL-OUTPUT))

30	<pre>(EQUAL L-IF-INTEGRATION 'NO) (EQUAL L-VOCL-IF 'NO) (EQUAL L-LIST-PROCESSING 'NO) (EQUAL L-ACCESS-CONFLICT 'NO) (EQUAL L-WORK-DIMENSION 'NO) (EQUAL L-LOOP-TAKE-IN 'NO) (EQUAL L-PARTITION 'NO) (EQUAL L-INTEGRATION 'NO) (EQUAL L-MONO 'NO) (EQUAL L-WRITE-DOWN 'NO) (EQUAL L-INTERCHANGE 'NO) (EQUAL L-UNROLLING 'NO) (EQUAL L-DIVISION 'NO) (EQUAL L-VOCL-SCALAR 'NO) (EQUAL L-VOCL-REPEAT 'NO) (EQUAL L-DELTA 'NO) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 2) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 4))</pre> <p>チューニング診断フラグが2か4であるのに、チューニング対策がみつからない。</p>	<pre>1 EXECUTE EXPRESSION: (PROGN (NOKS) (TERPRI TERMINAL-OUTPUT) (WAIT " ENTER KEY を押してください。"))</pre>
31	<pre>(EQUAL (@VALUE '@DATA 'QAJND) 'YES) (MEMBER 227 (@VALUE '@DATA 'QAJNDNNN)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))</pre> <p>ベクトル化メッセージ227が出力されている。</p>	<pre>(1) EXECUTE EXPRESSION: FORMAT " / N / C / N / N / C / N / N / C " " <メッセージ番号JND-227の意味> " " 入出力文は、ベクトル化できない。" " **メッセージ番号JND-227に対する処置 ** " (TERPRI TERMINAL-OUTPUT))</pre>
32	<pre>(MEMBER 227 (@VALUE 'ND-TUNNING-CONTROL 'J227-COMMENT)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))</pre> <p>ベクトル化メッセージ227に関するJ227-COMMENTのフラグがオンである。</p>	<pre>(1) EXECUTE EXPRESSION: FORMAT " / N / C / N / C " " デバッグ用に出力文など不要な文は、 コメントにするか、" " 削除するように修正してください。" (TERPRI TERMINAL-OUTPUT))</pre>
33	<pre>(MEMBER 227 (@VALUE 'ND-TUNNING-CONTROL 'J227-INDEPENDENT)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))</pre> <p>ベクトル化メッセージ227に関するJ227-INDEPENDENTのフラグがオンである。</p>	<pre>(1) EXECUTE EXPRESSION: FORMAT " / N / C / N / C " " 出力文中に計算式があれば、その計算は 独立した式で実行するように" " 修正して ください。また入出力を行う部分を独立し たDOループにする" " ことも検討してく ださい。" (TERPRI TERMINAL-OUTPUT))</pre>

34	<pre>(EQUAL (@VALUE '@DATA 'QAJND) 'YES) (MEMBBR 228 (@VALUE '@DATA 'QAJNDNNN)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))</pre> <p>ベクトル化メッセージ228 が出力されている。</p>	<p>(1) EXECUTE EXPRESSION: (FORMAT "/N/C/N/N/C/N/N/C" "<メッセージ番号JND-228 の意味>" "<外部手続き (CALL文, 関数引用は、ベクトル化できない。" " **メッセージ番号JND-228 に対する処置 **") (TERPRI TERMINAL-OUTPUT))</p>
35	<pre>(MEMBER 228 (@VALUE 'ND-TUNNING-CONTROL 'J228-EXPAND)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))</pre> <p>ベクトル化メッセージ228 に関するJ228-EXPAND のフラグがオンである。</p>	<p>(1) EXECUTE EXPRESSION: (FORMAT "/N/C/N/C/N/C" "<DOループの中に、その外部手続きを展開して下さい。展開は、""コンパイラのPIオプションを試して下さい。また文関数で""置換できるかも検討して下さい。") (TERPRI TERMINAL-OUTPUT))</p>
36	<pre>(MEMBER 228 (@VALUE 'ND-TUNNING-CONTROL 'J228-DIVIDE)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))</pre> <p>ベクトル化メッセージ228 に関するJ228-DIVIDE のフラグがオンである。</p>	<p>(1) EXECUTE EXPRESSION: (FORMAT "/N/C/N/C/N/C" "<外部手続きを利用している部分とその他の部分にDOループを分割し""その外部手続きの中にDOループを持ち込むことを検討して下さい。") (TERPRI TERMINAL-OUTPUT))</p>
37	<pre>(EQUAL (@VALUE '@DATA 'QAJND) 'YES) (MEMBER 229 (@VALUE '@DATA 'QAJNDNNN)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))</pre> <p>ベクトル化メッセージ229 が出力されている。</p>	<p>(1) EXECUTE EXPRESSION: (FORMAT "/N/C/N/N/C/N/N/C" "<メッセージ番号JND-229 の意味>" "<文字型データはベクトル化できない。" " **メッセージ番号JND-229 に対する処置 **") (TERPRI TERMINAL-OUTPUT))</p>
38	<pre>(MEMBER 229 (@VALUE 'ND-TUNNING-CONTROL 'J229-CHARACTER)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3))</pre> <p>ベクトル化メッセージ229 に関するJ229-CHARACTER のフラグがオンである。</p>	<p>(1) EXECUTE EXPRESSION: (FORMAT "/N/C" "<4バイト整数型または8バイト実数型への修正を検討して下さい。") (TERPRI TERMINAL-OUTPUT))</p>

39	(EQUAL (@VALUE '@DATA 'QAJND) 'YES) (MEMBER 231 (@VALUE '@DATA 'QAJNDNNN)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ231が出力されている。	(1) EXECUTE EXPRESSION: (FORMAT " /N /C /N /N /C /N /N /C " " <メッセージ番号JND-231の意味>" " ある配列の添字変数が回帰参照を引き起こす可能性がある." " **メッセージ番号JND-231に対する処置 **" (TERPRI TERMINAL-OUTPUT))
40	(EQUAL (@VALUE 'ND-TUNNING-CONTROL ' J231-DIVIDE) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ231に関するJ231-DIVIDEのフラグがオンである。	(1) EXECUTE EXPRESSION: (FORMAT " /N /C /N /C " " 回帰参照でベクトル化できない部分とその他のベクトル化できる部分" "にDOループを分割して下さい。") (TERPRI TERMINAL-OUTPUT))
41	(EQUAL (@VALUE 'ND-TUNNING-CONTROL ' J231-INTERCHANGE) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ231に関するJ231-INTERCHANGEのフラグがオンである。	(1) EXECUTE EXPRESSION: (FORMAT " /N /C " " DOループの入れ替えによる回帰参照の除去を検討して下さい。") (TERPRI TERMINAL-OUTPUT))
42	(EQUAL (@VALUE 'ND-TUNNING-CONTROL ' J231-VOCL-NOVREC) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ231に関するJ231-VOCL-NOVRECのフラグがオンである。	(1) EXECUTE EXPRESSION: (FORMAT " /N /C " " DO文の直前に* VOCL LOOP, NOVREC文を挿入してください。") (TERPRI TERMINAL-OUTPUT))
43	(EQUAL (@VALUE '@DATA 'QAJND) 'YES) (MEMBER 249 (@VALUE '@DATA 'QAJNDNNN)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ249が出力されている。	(1) EXECUTE EXPRESSION: (FORMAT " /N /C /N /N /C /N /N /C " " <メッセージ番号JND-249の意味>" " あるデータは、入出力・外部手続き・文字データで回帰参照を起こす可能性がある。 。") " **メッセージ番号JND-249に対する処置 **" (TERPRI TERMINAL-OUTPUT))

44	<pre>(MEMBER 249 (@VALUE 'ND-TUNNING-CONTROL 'J227-COMMENT) 'YES) (EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-OTHER) 'IO-END) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ249 に関するJ227-COMMENTの フラグがオンである。</pre>	<p>(1) EXECUTE EXPRESSION: (PROGN (FORMAT " N / C / N / C " " デバッグ用の出力文など必要な文は、 コメントにするか、" " 削除するように修 正してください。") (TERPRI TERMINAL-OUTPUT))</p>
45	<pre>(MEMBER 249 (@VALUE 'ND-TUNNING-CONTROL 'J227-INDEPENDENT) 'YES) (EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-OTHER) 'IO-END) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ249 に関するJ227-INDEPENDENTの フラグがオンである。</pre>	<p>(1) EXECUTE EXPRESSION: (PROGN (FORMAT " /N /C /N /C " " 出力文中に計算式があれば、その計算は 独立した式で実行するように" " 修正して ください。また入出力を行う部分を独立し たDOループにする" " ことも検討してくだ さい。") (TERPRI TERMINAL-OUTPUT))</p>
46	<pre>(MEMBER 249 (@VALUE 'ND-TUNNING-CONTROL 'J228-EXPAND) 'YES) (EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-OTHER) 'EXT-END) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ249 に関するJ228-EXPAND の フラグがオンである。</pre>	<p>(1) EXECUTE EXPRESSION: (PROGN (FORMAT " /N /C /N /C /N /C " " DOループの中に、その外部手続きを展開 して下さい。展開は、" " コンパイラのPI オプションを試して下さい。また文関数で " " 置換できるかも検討して下さい。") (TERPRI TERMINAL-OUTPUT))</p>
47	<pre>(MEMBER 249 (@VALUE 'ND-TUNNING-CONTROL 'J228-DIVIDE)) (EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-OTHER) 'EXT-END) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ249 に関するJ228-DIVIDE の フラグがオンである。</pre>	<p>(1) EXECUTE EXPRESSION: (PROGN (FORMAT " /N /C /N /C /N /C " " 外部手続きを利用している部分とその他の 部分にDOループを分割し" " その外部手 続きの中にDOループを持ち込むことを検討 して下さい。") (TERPRI TERMINAL-OUTPUT))</p>
48	<pre>(MEMBER 249 (@VALUE 'ND-TUNNING-CONTROL 'J229-CHARACTER) 'YES) (EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-OTHER) 'CHR) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ249 に関するJ229-CHARACTER のフラグがオンである。</pre>	<p>(1) EXECUTE EXPRESSION: (PROGN (FORMAT " /N /C " " 4バイト整数型または8バイト実数型へ の修正を検討して下さい。") (TERPRI TERMINAL-OUTPUT))</p>

49	<pre>(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-DIVIDE) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ249 に関するJ249-DIVIDE のフラグがオンである。</pre>	<pre>(1) EXECUTE EXPRESSION : (PROGN FORMAT " /N /C /N /C " " 回帰参照でベクトル化できない部分とそ の他のベクトル化できる部分" "にDOルー プを分割して下さい。") (TERPRI TERMINAL-OUTPUT))</pre>
50	<pre>(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-INTERCHANGE) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ249 に関するJ249-INTERCHAN GEのフラグがオンである。</pre>	<pre>(1) EXECUTE EXPRESSION : (PROGN FORMAT " /N /C " " DOループの入れ替えによる回帰参照の除 去を検討して下さい。") (TERPRI TERMINAL-OUTPUT))</pre>
51	<pre>(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-NOKS) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ249 に関する対策がない。</pre>	<pre>(1) EXECUTE EXPRESSION : (PROGN (NOKS) (TERPRI TERMINAL-OUTPUT) (WAIT " ENTER KEY を押して下さい。"))</pre>
52	<pre>(EQUAL (@VALUE '@DATA 'QAJND) 'YES) (MEMBER 250 (@VALUE '@DATA 'QAJNDNNN)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ250 が出力されている。</pre>	<pre>(1) EXECUTE EXPRESSION : FORMAT " /N /C /N /N /C /N /N /C " " <メッセージ番号JND-250 の意味> " " ある配列が、回帰参照を起こしている。 ") " **メッセージ番号JND-250 に対する処 置 **") (TERPRI TERMINAL-OUTPUT))</pre>
53	<pre>(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J250-DIVIDE) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ250 に関するJ250-DIVIDE のフラグがオンである。</pre>	<pre>(1) EXECUTE EXPRESSION : (PROGN FORMAT " /N /C /N /C " " 回帰参照でベクトル化できない部分とそ の他のベクトル化できる部分" "にDOルー プを分割して下さい。") (TERPRI TERMINAL-OUTPUT))</pre>
54	<pre>(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J250-INTERCHANGE) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ250 に関するJ250-INTERCHAN GEのフラグがオンである。</pre>	<pre>(1) EXECUTE EXPRESSION : (PROGN FORMAT " /N /C " " DOループの入れ替えによる回帰参照の除 去を検討して下さい。") (TERPRI TERMINAL-OUTPUT))</pre>

55	(EQUAL (@VALUE '@DATA 'QAJND) 'YES) (MEMBER 262 (@VALUE '@DATA 'QAJNDNNN)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ262 が出力されている。	(1) EXECUTE EXPRESSION: (FORMAT "/N/C/N/N/C/N/N/C" " <メッセージ番号JND-262 の意味>" " データの依存関係が回帰的なのでベクトル化できない部分がある。" " **メッセージ番号JND-262 に対する処置 **") (TERPRI TERMINAL-OUTPUT))
56	(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J262-INTERCHANGE) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ262 に関するJ262-INTERCHANGEのフラグがオンである。	(1) EXECUTE EXPRESSION: (PROGN (FORMAT "/N/C" " DOループの入れ替えによる回帰参照の除去を検討して下さい。") (TERPRI TERMINAL-OUTPUT))
57	(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J262-NOKS) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ262 に関する対策がない。	(1) EXECUTE EXPRESSION: (PROGN (NOKS) (TERPRI TERMINAL-OUTPUT) (WAIT " ENTER KEY を押して下さい。"))
58	(EQUAL (@VALUE '@DATA 'QAJND) 'YES) (MEMBER 277 (@VALUE '@DATA 'QAJNDNNN)) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ277 が出力されている。	(1) EXECUTE EXPRESSION: (FORMAT "/N/C/N/N/C/N/N/C" " <メッセージ番号JND-277 の意味>" " 外側ループで定義されたあるデータが内側ループ内で使われている。" " **メッセージ番号JND-277 に対する処置 **") (TERPRI TERMINAL-OUTPUT))
59	(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J277-WRITE-DOWN) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ277 に関して書き下しフラグがオンである。	(1) EXECUTE EXPRESSION: (PROGN (FORMAT "/N/C" " 最内ループの書き下しをデータを検討してください。") (TERPRI TERMINAL-OUTPUT))
60	(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J277-WORK) 'YES) (OR (= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) (= (@VALUE 'ND-TUNNING 'CONCLUSION) 3)) ベクトル化メッセージ277 に関して作業配列フラグがオンである。	(1) EXECUTE EXPRESSION: (PROGN (FORMAT "/N/C/N/C" " メッセージに出力されたデータを作業配列に格納し、その配列を使って" " アクセスすることを検討してください。") (TERPRI TERMINAL-OUTPUT))

61	<p>(NOT (EQUAL (@VALUE 'ND-TUNNING 'CONCLUSION) 3))</p> <p>ベクトル化診断とチューニング診断の両方がフラグが同時にオンになっていない。</p>	<p>(1) EXECUTE EXPRESSION: (PROGN (FORMAT "/N/N/N") (TERPRI TERMINAL-OUTPUT) (WAIT "診断はすべて終了しました。 ENTER KEY を押して下さい。"))</p> <p>(2) PROPOSE CHANGE TYPE : NULL EVENT : INIT NODE : 'ND-TUNNING-CONTROL 診断内容選択KSを発行する。</p>
62	<p>(EQUAL (@VALUE 'ND-TUNNING 'CONCLUSION) 3))</p> <p>ベクトル化診断とチューニング診断の両方がフラグが同時にオンになっている。</p>	<p>(1) EXECUTE EXPRESSION: (PROGN (FORMAT "/N/N/N") (TERPRI TERMINAL-OUTPUT) (WAIT "診断の1つが終了しました。 ENTER KEY を押して下さい。"))</p> <p>(2) PROPOSE CHANGE TYPE : NULL EVENT : INIT NODE : 'ND-TUNNING ATTRIBUTES : CONCLUSION = (PLUS (@VALUE 'ND-TUNNING 'CONCLUSION) 1)) チューニング診断を行う。</p>

KS name	KS-VECTOR (FORWARD)	ベクトル化診断制御KS
trigger	VECTORIZE	
strategy	SINGLE MANYTIMES	
LOCAL-VARIABLE	EXPRESSION	
L-DOLoop L-IFSTMT	(NTH 2 (@VALUE 'ND-TUNNING 'VECTOR-LEN)) (@VALUE '@DATA 'QAIIFSTMT)	
RULE NO.	L II S (条 件 部)	R H S (実 行 部)
1	(OR (AND (> L-DOLoop 7) (EQUAL L-IFSTMT 'YES)) (AND (> L-DOLoop 5) (EQUAL L-IFSTMT 'NO))) ベクトル長が 7 より大きく、 IF文がある場合または ベクトル長が 5 より大きく、 IF文がない場合。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : VMSG NODE : '@DATA ベクトル化メッセージ診断KSを発行する。
2	(OR (AND (<= L-DOLoop 7) (EQUAL L-IFSTMT 'YES)) (AND (<= L-DOLoop 5) (EQUAL L-IFSTMT 'NO))) ベクトル長が 7 以下で、 IF文がある場合または ベクトル長が 5 以下で、 IF文がない場合。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : TUNNING-END NODE : 'ND-TUNNING チューニング結果出力KSを発行する。 (1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL ATTRIBUTES : VOCL-SCALAR = 'YES * VOCL SCALAR 文の挿入フラグを YES にする
3	T 1, 2 以外	(1) EXECUTE EXPRESSION: (PROGN (NOKS) (TERPRI TERMINAL-OUTPUT) (WAIT " ENTER KEY を押して下さい。")

KS name	KS-VMSG (FORWARD)	ベクトル化メッセージ診断制御KS
trigger	VMSG	
strategy	MULTIPLE MANYTIMES	
LOCAL-VARIABLE	EXPRESSION	
L-VECTORIZE L-JNDNNN	(@VALUE '@DATA 'QAVECTORIZED) (@VALUE '@DATA 'QAJNDNNN)	
RULE NO.	L H S (条 件 部)	R H S (実 行 部)
1	(CAR (EXTRACT L-JNDNNN L-MSG)) メッセージ番号が現在あるKSで対処できる。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : TUNNING-CONTROL NODE : 'ND-TUNNING チューニング結果出力KSを発行する。
2	(OR (MEMBER 201 L-JNDNNN) (MEMBER 202 L-JNDNNN) (MEMBER 203 L-JNDNNN) (MEMBER 204 L-JNDNNN) (MEMBER 205 L-JNDNNN) (MEMBER 206 L-JNDNNN) (MEMBER 207 L-JNDNNN)) (EQUAL L-VECTORIZED 'NO) ベクトル化メッセージでベクトル部分があることが示されているが、チューニング診断フラグがオンになっていない。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : TUNNING NODE : '@DATA チューニングKSを発行する。
3	(MEMBER 227 L-JNDNNN) メッセージ番号に227が出力されている。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : JND227 NODE : '@DATA JND-227 のKSを発行する。
4	(MEMBER 228 L-JNDNNN) メッセージ番号に228が出力されている。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : JND228 NODE : '@DATA JND-228 のKSを発行する。
5	(MEMBER 229 L-JNDNNN) メッセージ番号に229が出力されている。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : JND229 NODE : '@DATA JND-229 のKSを発行する。
6	(MEMBER 231 L-JNDNNN) メッセージ番号に231が出力されている。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : JND231 NODE : '@DATA JND-231 のKSを発行する。

7	(MEMBER 249 L-JNDNNN) メッセージ番号に249が出力されている。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : JND249 NODE : '@DATA JND-249 の KSを発行する。
8	(MEMBER 250 L-JNDNNN) メッセージ番号に250が出力されている。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : JND250 NODE : '@DATA JND-250 の KSを発行する。
9	(MEMBER 262 L-JNDNNN) メッセージ番号に262が出力されている。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : JND262 NODE : '@DATA JND-262 の KSを発行する。
10	(MEMBER 273 L-JNDNNN) メッセージ番号に273が出力されている。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : JND273 NODE : '@DATA JND-273 の KSを発行する。
11	(MEMBER 277 L-JNDNNN) メッセージ番号に277が出力されている。	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : JND277 NODE : '@DATA JND-277 の KSを発行する。
12	(NULL (CAR (EXTRACT L-JNDNNN L-MSG))) (NULL (CADR (EXTRACT L-JNDNNN L-MSG))) メッセージが出力されたと答えたにもかかわらず メッセージの番号が入力されていない。	(1) EXECUTE EXPRESSION : (PROGN (ERASE) (FORMAT " /N /C /N /C " " < > < > < 診断不能 > < > < > " " 出力されたメッセージを入力してください。") (WAIT-ENTER)
13	(CADR (EXTRACT L-JNDNNN L-MSG)) メッセージ番号が現在あるKSで対処できない。	(1) EXECUTE EXPRESSION : (PROGN (ERASE) (FORMAT " /N /C /N /C /N /C /N /C / C /N " " < > < > < 診断不可能 > < > < > " " このコンパイラメッセージは、現在" " は考慮しておりません。申し訳ありませんが、" " メッセージの出力された例を" " 開発担当者までお知らせください。") (TERPRI TERMINAL-OUTPUT))

KS name	KS-JND227 (FORWARD)	JND-227 診断KS
trigger	JND227	
strategy	SINGLE MANYTIMES	

RULE NO.	L し (条 件 部)	H H S (実 行 部)
1	<pre>(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-OTHER) '10) (EQUAL (@VALUE '@DATA 'QA10#2-249) 'YES)</pre> <p>JND-249 で出力されたデータは、デバッグ用の文である。</p>	<pre>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J227-COMMENT= (CONS 249 (@VALUE 'ND-TUNNING-CONTROL 'J227-COMMENT)) J249-OTHER = '10-END JND-249 での診断をし、データをコメント化を指示するフラグをオンにする。</pre>
2	<pre>(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-OTHER) '10) (EQUAL (@VALUE '@DATA 'QA10#2-249) 'NO)</pre> <p>JND-249 で出力されたデータは、デバッグ用の文でない。</p>	<pre>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J227-INDEPENDENT = (CONS 249 (@VALUE 'ND-TUNNING-CONTRO L 'J227-INDEPENDENT)) J249-OTHER = '10-END JND-249 での診断をし、その部分の独立を指示するフラグをオンする。</pre>
3	<pre>(MEMBER 227 (@VALUE '@DATA 'QAJNDNNN)) (EQUAL (@VALUE '@DATA 'QA10#2-227) 'YES)</pre> <p>JND-227 で出力されたデータは、デバッグ用の文である。</p>	<pre>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J227-COMMENT= (CONS 227 (@VALUE 'ND-TUNNING-CONTROL 'J227-COMMENT)) JND-227 での診断をし、データのコメント化を指示するフラグをオンにする。</pre>
4	<pre>(MEMBER 227 (@VALUE '@DATA 'QAJNDNNN)) (EQUAL (@VALUE '@DATA 'QA10#2-227) 'NO)</pre> <p>JND-227 で出力されたデータは、デバッグ用の文でない。</p>	<pre>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J227-INDEPENDENT = (CONS 227 (@VALUE 'ND-TUNNING-CONTRO L 'J227-INDEPENDENT)) JND-227 での診断をし、その部分の独立を指示するフラグをオンする。</pre>

KS name	KS-JND228 (FORWARD)	JND-228 診断KS
trigger	JND228	
strategy	SINGLE MANYTIMES	

RULE NO.	L (条 件 部)	H S (実 行 部)
1	<p>(EQUAL (@VALUE 'ND-TUNNING-CONTROL' J249-OTHER) 'EXT')</p> <p>(EQUAL (@VALUE '@DATA 'QAEXT #2-249) 'YES')</p> <p>JND-249 で出力された外部手続きは、単純でかつステップ数が少ない。</p>	<p>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL' J228-EXPAND = (CONS 249 (@VALUE 'ND-TUNNING-CONTROL' J228-EXPAND)) J249-OTHER = 'EXT-END' JND-249 での診断をし、外部手続きの展開を指示するフラグをオンにする。</p>
2	<p>(EQUAL (@VALUE 'ND-TUNNING-CONTROL' J249-OTHER) 'EXT')</p> <p>(EQUAL (@VALUE '@DATA 'QAEXT #2-249) 'NO')</p> <p>JND-249 で出力された外部手続きは、単純でないかステップ数が多い。</p>	<p>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL' J228-DIVIDE = (CONS 249 (@VALUE 'ND-TUNNING-CONTROL' J228-DIVIDE)) J249-OTHER = 'EXT-END' JND-249 での診断をし、その部分の独立を指示するフラグをオンする。</p>
3	<p>(MEMBER 228 (@VALUE '@DATA 'QAJNDNNN))</p> <p>(EQUAL (@VALUE '@DATA 'QAEXT #2-228) 'YES')</p> <p>JND-228 で出力された外部手続きは、単純でかつステップ数が少ない。</p>	<p>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL' J228-EXPAND = (CONS 228 (@VALUE 'ND-TUNNING-CONTROL' J228-EXPAND)) JND-228 での診断をし、外部手続きの展開を指示するフラグをオンにする。</p>
4	<p>(MEMBER 228 (@VALUE '@DATA 'QAJNDNNN))</p> <p>(EQUAL (@VALUE '@DATA 'QAEXT #2-228) 'NO')</p> <p>JND-228 で出力された外部手続きは、単純でないかステップ数が多い。</p>	<p>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL' J228-DIVIDE = (CONS 228 (@VALUE 'ND-TUNNING-CONTROL' J228-DIVIDE)) JND-228 での診断をし、その部分の独立を指示するフラグをオンする。</p>

KS name	KS-JND229 (FORWARD)	JND-229 診断KS
trigger	JND229	
strategy	SINGLE MANYTIMES	

RULE NO.	L H S (条 件 部)	R H S (実 行 部)
1	<p>(MEMBER 229 (@VALUE '@DATA 'QAJNDNNN))</p> <p>JND-229 が出力された。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J229-CHARACTER = (CONS 229 (@VALUE 'ND-TUNNING-CONTROL 'J229-CHARACTER)) JND-229 での文字データの型変換を指示する フラグをオンにする。</p>
2	<p>(EQUAL (@VALUE 'ND-TUNNING-CONTROL 'J249-OTHER) 'CHR)</p> <p>JND-249 で出力されたデータは、文字型データである。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J229-CHARACTER = (CONS 249 (@VALUE 'ND-TUNNING-CONTROL 'J229-CHARACTER)) JND-229 での文字データの型変換を指示する フラグをオンにする。</p>

KS name	KS-JND231 (FORWARD)	JND-231 診断KS
trigger	JND231	
strategy	SINGLE MANYTIMES	
LOCAL-VARIABLE	EXPRESSION	
L-REC L-MULT	(ASK-REC 'JND-231) (@VALUE 'ND-TUNNING 'QA-MULTIPLEX)	
RULE NO.	L H S (条 件 部)	R H S (実 行 部)
1	(EQUAL L-REC 'YES) JND-231 で出力されたデータは、回帰参照になっている。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J231-DIVIDE = 'YES 回帰参照になる部分とならない部分との分割を指示するフラグをオンにする。
2	(EQUAL L-REC 'YES) (>= L-MULT 2) JND-231 で出力されたデータは、回帰参照になっていて、かつ多重DOループである。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J231-DIVIDE = 'YES DOループの入れ替えによる回帰参照除去を指示するフラグをオンにする。
3	(EQUAL L-REC 'NO) JND-231 で出力されたデータは、回帰参照になっていない。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J231-VOCL-NOVREC = 'YES * VOCL NOVREC 文の挿入を指示するフラグをオンにする。

KS name	KS-JND249 (FORWARD)		JND-249 診断 KS
trigger	JND249		
strategy	SINGLE MANYTIMES		
LOCAL-VARIABLE	EXPRESSION		
L-REC L-MULT	(ASK-REC 'JND-249) (@VALUE 'ND-TUNNING 'QA-MULTIPLEX)		
RULE NO.	L H S (条 件 部)	R H S (実 行 部)	
1	(EQUAL L-REC 'YES) JND-249 で出力されたデータは、回帰参照になっている。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J249-DIVIDE = 'YES 回帰参照になる部分とならない部分との分割を指示するフラグをオンにする。	
2	(EQUAL L-REC 'YES) (>= L-MULT 2) JND-249 で出力されたデータは、回帰参照になっていて、かつ多重DOループである。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J249-INTERCHANGE = 'YES DOループの入れ替えによる回帰参照除去を指示するフラグをオンにする。	
3	(EQUAL L-REC 'NO) (EQUAL (@VALUE '@DATA 'QAI0#1) 'YES) JND-249 で出力されたデータは、回帰参照になっていない入出力文である。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : JND227 NODE : 'ND-TUNNING-CONTROL J249-OTHER = 'IO 入出力文のベクトル化に関するKS (JND227) を発行する。	
4	(EQUAL L-REC 'NO) (EQUAL (@VALUE '@DATA 'QAEXT #1) 'YES) JND-249 で出力されたデータは、回帰参照になっていない外部手続きである。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : JND228 NODE : 'ND-TUNNING-CONTROL J249-OTHER = 'EXT 外部手続きのベクトル化に関するKS (JND228) を発行する。	
5	(EQUAL L-REC 'NO) (EQUAL (@VALUE '@DATA 'QACHR #1) 'YES) JND-249 で出力されたデータは、回帰参照にならない文字型データである。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : JND229 NODE : 'ND-TUNNING-CONTROL J249-OTHER = 'CHR 文字型データのベクトル化のに関するKS (JND229) を発行する。	
6	T 1 ~ 5 以外	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J249-NOKS = 'YES KS不足を示すフラグをオンにする。	

KS name	KS-JND250 (FORWARD)	JND-250 診断KS
trigger	JND250	
strategy	MULTIPLE MANYTIMES	
LOCAL-VARIABLE	EXPRESSION	
L-MULT	(@VALUE 'ND-TUNNING' 'QA-MULTIPLEX)	
RULE NO.	L (条) H 件 S 部)	R (実) H 行 S 部)
1	T 常に	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J250-DIVIDE = 'YES 回帰参照になる部分とならない部分との分割を指示するフラグをオンにする。
2	(>= L-MULT 2) DOループの多重度が2以上である。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J250-INTERCHANGE = 'YES DOループの入れ替えによる回帰参照除去を指示するフラグをオンにする。

KS name	KS-JND262 (FORWARD)	JND-262 診断KS
trigger	JND262	
strategy	SINGLE MANYTIMES	
LOCAL-VARIABLE	EXPRESSION	
L-MULT	(@VALUE 'ND-TUNNING' QA-MULTIPLEX)	
RULE NO.	L II S (条 件 部)	R H S (実 行 部)
1	(>= L-MULT 2) DOループの多重度が2以上である。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J262-INTERCHANGE = 'YES DOループの入れ替えによる回帰参照除去を指示するフラグをオンにする。
2 T	1以外	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J262-NOKS = 'YES KS不足のフラグをオンにする。

KS name	KS-JND273 (FORWARD)		JND-273 診断KS
trigger	JND273		
strategy	SINGLE MANYTIMES		
LOCAL-VARIABLE	EXPRESSION		
L-REC L-MULT	(ASK-REC 'JND-273) (@VALUE 'ND-TUNNING 'QA-MULTIPLEX)		
RULE NO.	L (条) H (件) S (部)	R (実) H (行) S (部)	
1	(EQUAL L-REC 'YES) JND-273 で出力されたデータは、回帰参照になっている。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J273-DIVIDE = 'YES 回帰参照になる部分とならない部分との分割を指示するフラグをオンにする。	
2	(EQUAL L-REC 'YES) (>= L-MULT 2) JND-273 で出力されたデータは、回帰参照になっていて、かつ多重DOループである。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J273-INTERCHANGE = 'YES DOループの入れ替えによる回帰参照除去を指示するフラグをオンにする。	
3	(EQUAL L-REC 'NO) JND-273 で出力されたデータは、回帰参照になっていない。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : JND227 NODE : 'ND-TUNNING-CONTROL J273-VOCL-NOVREC = 'YES * VOCL NOVREC 文の挿入を指示するフラグをオンにする。	
4	(= (@VALUE 'ND-TUNNING 'CONCLUSION) 1) ベクトル化診断のみ行われている。	(1) PROPOSE CHANGE TYPE: NULL EVENT NAME : TUNING NODE : 'ND-TUNNING チューニング診断も行うために、チューニングKSを発行する。	

KS name	KS-JND277 (FORWARD)	JND-277 診断KS
Trigger	JND277	
strategy	SINGLE MANYTIMES	

RULE NO.	L (条)	H (件)	S (部)	R (実)	H (行)	S (部)
1	(> (@VALUE 'ND-TUNNING 'QA-MULTIPLEX) 1) (<= (NTH 2 (CAR (LAST (@VALUE 'ND-TUNNING 'LOOP-LEN)))) 7) (@VALUE 'ND-TUNNING 'CHG-VAR) (OR (EQUAL (NTH 3 (CAR (LAST (@VALUE 'ND-TUNNING 'LOOP-LEN)))) 1) (EQUAL (NTH 3 (CAR (LAST (@VALUE 'ND-TUNNING 'LOOP-LEN)))) 1.0)) 最内DOループの繰り返し回数が7より小さくて、確 信度が1である多重DOループである。			(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J277-WRITE-DOWN = 'YES 最内DOループの展開を指示するフラグをオ ンにする。		
2	T 1以外			(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-TUNNING-CONTROL J277-WORK = 'YES 作業配列を使って外側ループで定義された 変数を内側ループでアクセスするフラグを オンにする。		

KS name	KS-ABEND (FORWARD)	異常終了処理制御KS
trigger	ABEND	
strategy	MULTIPLE MANYTIMES	
RULE NO.	L (条件部)	H (実行部)
1	(EQUAL (@VALUE 'CONTROL-ABEND 'FIRST) 'YES) 異常終了処理制御KSは、はじめて通過する。	(1) EXECUTE EXPRESSION : (PROGN (ERASE) (FORMAT "/N/C/N" "使用方法は、以下の通りです。") (FORMAT "/N/C/N/C/N" "1. 異常終了した時のエラートレースおよびソースプログラムを用意して下さい。") (FORMAT "/N/C/N/C/N/N/ N/N/N/N/N/N/N/N" "尚、質問には、特に指定の無い限り、'YES'または'NO'で答えてください。") (2) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : 'CONTROL-ABEND ATTRIBUTES : FIRST = 'NO 異常終了処理制御KSの初期フラグをNOにする。
2	(EQUAL (@VALUE 'CONTROL-ABEND 'FIRST) 'NO) 異常終了処理制御KSは、通過したことがある。	(1) EXECUTE EXPRESSION : (PROGN (CLEAR-NODE 'ND-ABEND) (CLEAR-NODE 'ND-ABEND-C6)) BBのNODEのND-ABEND, ND-ABEND-C6をクリアする。
3	(EQUAL (@VALUE 'ND-ABEND 'SCALAR) 'NO) スカラー処理して、正常終了していない。	(1) EXECUTE EXPRESSION : (PROGN (ERASE) (FORMAT "/N/C/N/N/C/N" "スカラー処理で正常に終了するようにして下さい。") (TERPRI TERMINAL-OUTPUT) (WAIT "ENTER KEY を押して下さい。") (2) PROPOSE CHANGE TYPE: NULL EVENT NAME: INIT NODE : 'CONTROL-TOTAL 診断内容選択KSを発行する。

4	<pre>(EQUAL (@VALUE 'ND-ABEND 'SCALAR) 'YES) (EQUAL (@VALUE 'ND-ABEND 'CODE) 'C6)</pre>	<p>(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : ABEND-C6 NODE : 'CONTROL-ABEND 異常終了コード240-0C6 の診断KSを発行する。</p> <p>(2) PROPOSE CHANGE TYPE : NULL EVENT NAME : KENSAKU-C6 NODE : 'CONTROL-ABEND 異常終了コード240-0C6 の検索KSを発行する。</p>
5	<pre>(EQUAL (@VALUE 'ND-ABEND 'SCALAR) 'YES) (EQUAL (@VALUE 'ND-ABEND 'CODE) 'C7)</pre>	<p>(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : ABEND-C7 NODE : 'CONTROL-ABEND 異常終了コード240-0C7 の診断KSを発行する。</p>

5	(EQUAL T3 'YES) OC6 対策 3 のフラグがYES である。	(1) EXECUTE EXPRESSION : (FORMAT " /N /C /N /C /N /C / N " " 実引数の 4 バイトの配列と 8 バイトの変 数を" " EQUIVALENCE 文で結合してデー タ" " が 8 バイト境界上にくるように修正 してください。")
6	T	(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : INIT NODE : 'CONTROL-TOTAL 診断内容選択KSを発行する。 (2) EXECUTE EXPRESSION : (PROGN (TERPRI TERMINAL-OUTPUT) (WAIT " ENTER KEY を押してください。 ")) 画面消去をWAITする。

KS name	KS-KENSAKU-C6 (BACKCHAIN)	C6診断KS
trigger	KENSAKU-C6	
strategy	MULTIPLE 'TAISAKU	

RULE NO.	L (条) H (件) S (部)	R (実) H (行) S (部)
1	(EQUAL (@VALUE 'ND-ABEND-C6 'SYOUJYOU-1) 'YES) OC6について症状1があらわれている 症状1は、エラーのおきたDOループには、*VOCL REPEAT文があり、かつ実際の繰り返し回数より小さい。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY LEVEL : LV-ABEND-C6 ATTRIBUTES : TAISAKU-1 = 'YES TAISAKU-2 = 'NO TAISAKU-3 = 'NO 対策1のフラグをYESにする。
2	(EQUAL (@VALUE 'ND-ABEND-C6 'SYOUJYOU-1) 'NO) (EQUAL (@VALUE 'ND-ABEND-C6 'SYOUJYOU-2) 'YES) (EQUAL (@VALUE 'ND-ABEND-C6 'SYOUJYOU-3) 'YES) OC6について症状2と症状3が同時にあらわれている。 症状2は、エラーのおきたルーチンの仮引数の中で、8バイトで宣言され、実引数では4バイトで宣言されている。症状3は、エラールーチンの実引数には8バイトの配列とEQUIVALENCEをきっているものがある。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY LEVEL : LV-ABEND-C6 ATTRIBUTES : TAISAKU-1 = 'NO TAISAKU-2 = 'YES TAISAKU-3 = 'NO 対策2のフラグをYESにする。
3	(EQUAL (@VALUE 'ND-ABEND-C6 'SYOUJYOU-1) 'NO) (EQUAL (@VALUE 'ND-ABEND-C6 'SYOUJYOU-2) 'YES) (EQUAL (@VALUE 'ND-ABEND-C6 'SYOUJYOU-3) 'NO) OC6について症状2があらわれている。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY LEVEL : LV-ABEND-C6 ATTRIBUTES : TAISAKU-1 = 'NO TAISAKU-2 = 'NO TAISAKU-3 = 'YES 対策3のフラグをYESにする。
4	(EQUAL (@VALUE 'ND-ABEND-C6 'SYOUJYOU-1) 'NO) (EQUAL (@VALUE 'ND-ABEND-C6 'SYOUJYOU-2) 'NO) 症状1も症状2もあらわせていない。	(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY LEVEL : LV-ABEND-C6 ATTRIBUTES : TAISAKU-1 = 'NO TAISAKU-2 = 'NO TAISAKU-3 = 'NO すべてのフラグをNOにする。

5	<pre>(OR (EQUAL (@VALUEBC 'ND-ABEND-C6 'TAISAKU-1) 'YES) T) (OR (EQUAL (@VALUEBC 'ND-ABEND-C6 'TAISAKU-2) 'YES) T) (OR (EQUAL (@VALUEBC 'ND-ABEND-C6 'TAISAKU-3) 'YES) T)</pre> <p>いずれかの対策が成立した。</p>	<pre>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY LEVEL : LV-ABEND-C6 ATTRIBUTES : TAISAKU = 'YES</pre> <p>対策の探索ができたことを示すフラグがオンになった。</p>
6	<pre>(NOT (EQUAL (@VALUEBC 'ND-ABEND-C6 'TAISAKU-1) 'YES)) (NOT (EQUAL (@VALUEBC 'ND-ABEND-C6 'TAISAKU-2) 'YES)) (NOT (EQUAL (@VALUEBC 'ND-ABEND-C6 'TAISAKU-3) 'YES))</pre> <p>いずれかの対策も成立しなかった。</p>	<pre>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY LEVEL : LV-ABEND-C6 ATTRIBUTES : TAISAKU = 'NO</pre> <p>対策の探索ができなかったことを示すフラグがオンになった。</p>

KS name	KS-ABEND-C7 (FORWARD)		C7診断制御KS
trigger	ABEND-C7		
strategy	MULTIPLE MANYTIMES		
LOCAL-VARIABLE	EXPRESSION		
L-Q	(@VALUE 'ND-ABEND-C7 'QUEUE)		
RULE NO.	L (条)	H (件)	S (部)
1	T		<p>(1) EXECUTE EXPRESSION : (ERASE) 画面消去をする。</p>
2	(EQUAL (@VALUE 'ND-ABEND-C7 'FIRST) 'YES) OC7 診断KSを通過するのは、初めてである。		<p>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-ABEND-C7 ATTRIBUTES : QUEUE = (LIST (LIST (RIND "異常終了したルーチンの名前を入力して 下さい。 ==>") (RIND "異常終了したベクトル演算を行っている データの名前を入力して下さい。 ==>") (RIND "異常終了したデータのタイプを入力して 下さい。 (R * 4, R * 8) ==>") 'NO-RNAME 'NO-DNAME 'NO-DTYPE)) FIRST = 'NO</p>
3	(@VALUE 'ND-ABEND-C7 'QUEUE) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'NO) (EQUAL (@VALUE 'ND-ABEND-C7 'KENSAKU-FIRST) 'NO) OC7 検索キューが空でなく、非正規化データも検出 されていない。また、このKSの通過も初めてでない 。		<p>(1) EXECUTE EXPRESSION : (TERPRI TERMINAL-OUTPUT)</p> <p>(2) PROPOSE CHANGE TYPE: ADD EVENT NAME : KENSAKU-C7 NODE : LV-KENSAKU-C7 ATTRIBUTES : RNAME = (NTH 0 (CAR (OBJ-REP (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND- C7 'QUEUE)))) DNAME = (NTH 1 (CAR (OBJ-REP (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND- C7 'QUEUE)))) DTYPE = (NTH 2 (CAR (OBJ-REP (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND- C7 'QUEUE)))))</p>

	<pre> DOWN-RNAME= (NTH 3 (CAR (OBJ-REF (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND- C7 'QUEUE)))) DOWN-DNAME= (NTH 4 (CAR (OBJ-REF (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND- C7 'QUEUE)))) DOWN-DTYPE= (NTH 5 (CAR (OBJ-REF (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND- C7 'QUEUE)))) LINKS : DOWN-ROUTINE = (@VALUE 'ND-ABEND-C7 'NODE-NAME) 検索対象データに関する情報(対象ルーチン名, 対象データ名, 対象データタイプ, 対象ルーチンの上位ルーチン名, 上位ルーチンでのデータ名, データタイプ, 下位ルーチンとのリンクをセットする。 </pre>
	<p>(3) PROPOSE</p> <pre> CHANGE TYPE: SUPERSEDe EVENT NAME : DUMMY NODE : 'ND-ABEND-C7 ATTRIBUTES : NODE-NAME = @NEWNODE QUEUE = (CDR (OBJ-REF (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND-C7 'QUEUE)))) </pre>
4	<p>(@VALUE 'ND-ABEND-C7 'QUEUE)</p> <p>(EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) , NO)</p> <p>(EQUAL (@VALUE 'ND-ABEND-C7 'KENSAKU-FIRST)) 'YES)</p> <p>OC7 検索キューが空でなく, 非正規化データも検出されていない。また, このKSの通過も初めてでない。</p> <p>(1) EXECUTE EXPRESSION : (TERPRI TERMINAL-OUTPUT)</p> <p>(2) PROPOSE</p> <pre> CHANGE TYPE: ADD EVENT NAME : KENSAKU-C7 NODE : LV-KENSAKU-C7 ATTRIBUTES : RNAME = (NTH 0 (CAR (OBJ-REF (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND- C7 'QUEUE)))) DNAME = (NTH 1 (CAR (OBJ-REF (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND- C7 'QUEUE)))) DTYPE = (NTH 2 (CAR (OBJ-REF (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND- C7 'QUEUE)))) DOWN-RNAME= (NTH 3 (CAR (OBJ-REF (@VALUE 'ND-ABEND-C7 'QUEUE) (CAAR (@VALUE 'ND-ABEND- C7 'QUEUE)))) </pre>

```

DOWN-DNAME = (NTH 4
  (CAR (OBJ-REF (@VALUE 'ND-ABEND-C7
    'QUEUE) (CAAR (@VALUE 'ND-ABEND-
    C7 'QUEUE) ) ) )
DOWN-DTYPE = (NTH 5
  (CAR (OBJ-REF (@VALUE 'ND-ABEND-C7
    'QUEUE) (CAAR (@VALUE 'ND-ABEND-
    C7 'QUEUE) ) ) )
LINKS :
DOWN-ROUTINE =
  (@VALUE 'ND-ABEND-C7 'NODE-NAME)
検索対象データに関する情報(対象ルーチン名, 対象データ名, 対象データタイプ,
対象ルーチンの上位ルーチン名, 上位ルーチンでのデータ名, データタイプ, 下位ルーチンとのリンクをセットする。

```

```

(3) PROPOSE
CHANGE TYPE: SUPERSEDE
EVENT NAME : DUMMY
NODE       : 'ND-ABEND-C7
ATTRIBUTES :
KENSAKU-FIRST = 'NO
NODE-NAME     = @NEWNODE
QUEUE        = (CDR (OBJ-REF
  (@VALUE 'ND-ABEND-C7 'QUEUE)
  (CAAR (@VALUE 'ND-ABEND-C7 'QUEUE) ) ) )

```

KS name	KS-KENSAKU-C7 (FORWARD)	C7探索制御KS
trigger	KENSAKU-C7	
strategy	MULTIPLE MANYTIMES	
LOCAL-VARIABLE	EXPRESSION	
L-RNAME L-DNAME L-D-RNAME	(@VALUE '@NEWNODE 'RNAME) (@VALUE '@NEWNODE 'DNAME) (@VALUE '@NEWNODE 'DOWN-RNAME)	
RULE NO.	L H S (条 件 部)	R H S (実 行 部)
1 T		(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : @NEWNODE ATTRIBUTES : DEF-VALUE= (ASK-DEF-VALUE L-RNAME L-DNAME L-D-RNAME) EQV-EXIST= (ASK #3 (KSTRING-APPEND (STRING L-RNAME) "の中で" (STRING L-DNAME) "と他の変数・配 列が, ") (KSTRING-APPEND " EQUIVALE NCE 文で結合され, かつ" (STRING L-DNA ME)) "より新しく定義されていますか ?") 対象データの定義値を特定する。
2 (EQUAL (@VALUE @NEWNODE 'EQV-EXIST) 'YE S) 対象データがEQUIVALENCE で他の変数と結合されて いる。		(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : @NEWNODE ATTRIBUTES : EQV-DNAME= (ASKNAME "結合されているデータの名前を入力して ください。") EQV-DTYPE= (ASKTYPE "結合されているデータのタイプを入力し てください。") EQV-DEF-VALUE = (ASK-DEF-VALUE L-RNAME (@VALUE @NEWNODE 'EQV-DNAME) L-D-RNAME) EQUIVALENCE 結合データの定義値を特定す る。

3	<pre>(EQUAL (@VALUE @NEWNODE 'EQV-EXIST) 'YES) (EQUAL (@VALUE @NEWNODE 'EQV-DEF-VALUE) 3) (EQUAL (SEARCH-ABNORMAL (@VALUE @NEWNODE 'EQV-DTYPE) (@VALUE @NEWNODE 'DTYPE)) 'YES)</pre> <p>対象データがEQUIVALENCE で他の変数と結合されていて、その定義値タイプが3 で、対象データとの整合性（非正規化データのチェック）の結果、非正規化データである。</p>	<p>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES: HI-SEIKIKA = 'YES TAISAKU = 6 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME ERROR-EQV-DNAME= (@VALUE @NEWNODE 'EQV-DNAME) エラーイプ6 で非正規化データ発見。結果出力KSを発行する。</p>
4	<pre>(EQUAL (@VALUE @NEWNODE 'EQV-EXIST) 'YES) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'NO)</pre> <p>対象データがEQUIVALENCE で他の変数と結合されているが、非正規化データでない。</p>	<p>(1) PROPOSE CHANGE TYPE: NULL EVENT NAME: KENSAKU2-C7 NODE : @NEWNODE ATTRIBUTES: EQUIVALENCE データの検索KSを発行する。</p>
5	<pre>(EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'NO)</pre> <p>非正規化データが発見されていない。</p>	<p>(1) PROPOSE CHANGE TYPE: NULL EVENT NAME: KENSAKU1-C7 NODE : @NEWNODE ATTRIBUTE: 対象データの検索KSを発行する。</p>

KS name	KS-KENSAKU1-C7 (FORWARD)	C7元データ探索KS
trigger	KENSAKU1-C7	
strategy	MULTIPLE MANYTIMES	
LOCAL-VARIABLE	EXPRESSION	
L-Q L-Q-WORK L-DEF L-RNAME L-DNAME L-DTYPE L-D-RNAME L-D-DNAME L-D-DTYPE	(@VALUE 'ND-ABEND-C7' 'QUEUE) NIL (@VALUE '@FOCUSNODE 'DEF-VALUE) (@VALUE '@FOCUSNODE 'RNAME) (@VALUE '@FOCUSNODE 'DNAME) (@VALUE '@FOCUSNODE 'DTYPE) (@VALUE '@FOCUSNODE 'DOWN-RNAME) (@VALUE '@FOCUSNODE 'DOWN-DNAME) (@VALUE '@FOCUSNODE 'DOWN-DTYPE)	
RULE NO.	L (条 件 部)	H (実 行 部)
1	(EQUAL L-DEF 1) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'NO)	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : HI-SEIKIKA = 'YES TAISAKU = 1 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME エラータイプ1で非正規化データ発見。結果出力KSを発行する。
2	(EQUAL L-DEF 3) (EQUAL (SEARCH-ABNORMAL L-DTYPE L-D-DTYPE) 'YES) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'NO) 定義値タイプが3で、対象データと下位ルーチンでのデータが非正規化データの原因になっている。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : HI-SEIKIKA = 'YES TAISAKU = 2 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME ERROR-DTYPE=L-DTYPE ERROR-D-RNAME=L-D-RNAME ERROR-D-DNAME=L-D-DNAME ERROR-D-DTYPE=L-D-DTYPE エラータイプ2で非正規化データ発見。結果出力KSを発行する。

3	<pre>(OR (EQUAL L-DEF 4) (EQUAL L-DEF 5)) (EQUAL (@VALUE @FOCUSNODE 'COMMON-ASK) 'NO) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0)</pre> <p>定義値タイプが4または、5であるとき、COMMON変数の先頭アドレスが一致していない。</p>	<p>(1) PROPOSE</p> <p>CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES: HI-SEIKIKA = 'YES TAISAKU = 3 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME ERROR-DTYPE=L-DTYPE ERROR-D-RNAME=L-D-RNAME ERROR-D-DNAME=L-D-DNAME ERROR-D-DTYPE=L-D-DTYPE</p> <p>エラータイプ3で非正規化データ発見。結果出力KSを発行する。</p>
4	<pre>(EQUAL L-DEF 8) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) (OR (EQUAL (@VALUE @FOCUSNODE 'DOWN-ASK) 1) (AND (EQUAL (@VALUE @FOCUSNODE 'DOWN-ASK) 3) (EQUAL (SEARCH-ABNORMAL L-DTYPE L-D-DTYPE) 'YES))) (@VALUE @FOCUSNODE 'DOWN-DEF-RNAME)</pre> <p>定義値タイプが8で、下位ルーチンでの定義値が1かまたは、定義値が3で非正規化データである。</p>	<p>(1) PROPOSE</p> <p>CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES: HI-SEIKIKA = 'YES TAISAKU = 4 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME ERROR-DTYPE=L-DTYPE ERROR-D-RNAME= (@VALUE @FOCUSNODE 'DOWN-DEF-RNAME)</p> <p>エラータイプ4で非正規化データ発見。結果出力KSを発行する。</p>
5	<pre>(EQUAL L-DEF 9) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0)</pre> <p>定義値タイプが9である。</p>	<p>(1) PROPOSE</p> <p>CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES: HI-SEIKIKA = 'YES TAISAKU = 5 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME ERROR-DTYPE=L-DTYPE ERROR-D-RNAME=L-D-RNAME ERROR-D-DNAME=L-D-DNAME ERROR-D-DTYPE=L-D-DTYPE</p> <p>エラータイプ5で非正規化データ発見。結果出力KSを発行する。</p>
6	<pre>(EQUAL L-DEF 2) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0)</pre> <p>定義値タイプが2である。</p>	<p>(1) PROPOSE</p> <p>CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : @FOCUSNODE ATTRIBUTES: BACK = 2 EQV-BACK = 0</p> <p>バック・トラックタイプを2にする。</p>

7	<pre>(EQUAL L-DEF 3) (EQUAL (SEARCH-ABNORMAL L-D-TYPE L-D-TYPE) 'NO) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) 定義値タイプが3である。</pre>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : @FOCUSNODE ATTRIBUTES : BACK = 3 EQV-BACK = 0 パック・トラックタイプを3にする。</p>
8	<pre>(EQUAL L-DEF 8) (EQUAL (@VALUE @FOCUSNODE 'DOWN-ASK) 2) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) 定義値タイプが8であり、かつ下位ルーチンの定義値タイプが2である。</pre>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : @FOCUSNODE ATTRIBUTES : BACK = 2 EQV-BACK = 0 パック・トラックタイプを2にする。</p>
9	<pre>(EQUAL L-DEF 8) (EQUAL (@VALUE @FOCUSNODE 'DOWN-ASK) 3) (EQUAL (SEARCH-ABNORMAL L-D-TYPE L-D-TYPE) 'NO) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) 定義値タイプが8であり、かつ下位ルーチンの定義値タイプが3で非正規化データでない。</pre>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : @FOCUSNODE ATTRIBUTES : BACK = 3 EQV-BACK = 0 パック・トラックタイプを3にする。</p>
10	<pre>(NOT (@VALUE 'ND-ABEND-C7 'BACK-FLG)) (EQUAL L-DEF 4) (EQUAL (@VALUE @FOCUSNODE 'COMMON-ASK) 'Y ES) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) 定義値タイプが4であり、かつパック・トラック中でなく、COMMON変数の先頭アドレスが一致している</pre>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : ABEND-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : QUEUE = (ASK-QUEUE-COMMON L-Q L-RNAME L-DNAME L-DTYPE 'NO) COMMON変数の間接定義値をセットする。</p>
11	<pre>(NOT (@VALUE 'ND-ABEND-C7 'BACK-FLG)) (EQUAL L-DEF 5) (EQUAL (@VALUE @FOCUSNODE 'COMMON-ASK) 'Y ES) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) 定義値タイプが5であり、かつパック・トラック中でなく、COMMON変数の先頭アドレスが一致している</pre>	<p>(1) EXECUTE EXPRESSION : (SETQ L-Q-WORK (ASK-QUEUE-COMMON L-Q L-RNAME L-DNAME L-DTYPE 'YES) COMMON変数の直接定義値を作業変数にセットする。</p> <p>(2) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : ABEND-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : QUEUE = (CDR L-Q-WORK) さらに上位ルーチンを探索するためのKSを発行する。</p> <p>(3) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : @FOCUSNODE ATTRIBUTES : K-DEF-VALUE = (CAR L- Q-WORK) 間接定義値タイプを保存する。</p>

12	<pre>(NOT (@VALUE 'ND-ABEND-C7 'BACK-FLG)) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) (EQUAL L-DEF 6)</pre> <p>定義値タイプが6であり、かつバック・トラック中でない。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : ABEND-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : QUEUE = (ASK-QUEUE L-Q L-RNAME L-DNAME L-DTYPE 'NO) 対象データの実引数の定義値タイプをセットする。</p>
13	<pre>(NOT (@VALUE 'ND-ABEND-C7 'BACK-FLG)) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) (EQUAL L-DEF 7)</pre> <p>定義値タイプが7であり、かつバック・トラック中でない。</p>	<p>(1) EXECUTE EXPRESSION : (SETQ L-Q-WORK (ASK-QUEUE L-Q L-RNAME L-DNAME L-DTYPE 'YES)) 対象データの実引数の定義値タイプを作業変数にセットする。</p> <p>(2) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : ABEND-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : QUEUE = (CDR L-Q-WORK))さらに上位ルーチンを探索するためのKSを発行する。</p> <p>(3) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : @FOCUSNODE ATTRIBUTES : K-DEF-VALUE = (CAR L-Q-WORK) 間接定義値タイプを保存する。</p>
14	<pre>(EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) (@VALUE 'ND-ABEND-C7 'QUEUE) (@VALUE 'ND-ABEND-C7 'BACK-FLG)</pre> <p>バック・トラック中で、かつまだ探索キューが空でない。</p>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : ABEND-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : NODE-NAME = @FOCUSNODE BACK-FLG = NIL</p>
15	<pre>(EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) (NOT (EQUAL (@VALUE @FOCUSNODE 'BACK) 0)) (NOT (EQUAL (@VALUES @FOCUSNODE 'DOWN-ROUTNE) 'ND-ABEND-C7))</pre> <p>バック・トラックタイプが指定(0以外)されていて、エラールーチンまでバック・トラックしていない。</p>	<p>(1) PROPOSE CHANGE TYPE : NULL EVENT NAME : BACK-C7 NODE : @FOCUSNODE</p> <p>(2) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : 'ND-ABEND-C7 ATTRIBUTES : BACK-FLG=T</p>

KS name	KS-KENSAKU2-C7 (FORWARD)	C7EQUIVALENCE データ検索KS
trigger	KENSAKU2-C7	
strategy	MULTIPLE MANYTIMES	
LOCAL-VARIABLE	EXPRESSION	
L-Q L-Q-WORK L-DEF L-RNAME L-DNAME L-DTYPE L-D-RNAME L-D-DNAME L-D-DTYPE	(@VALUE 'ND-ABEND-C7' 'QUEUE) NIL (@VALUE '@FOCUSNODE' 'EQV-DEF-VALUE') (@VALUE '@FOCUSNODE' 'RNAME') (@VALUE '@FOCUSNODE' 'EQV-DNAME') (@VALUE '@FOCUSNODE' 'EQV-DTYPE') (@VALUE '@FOCUSNODE' 'DOWN-RNAME') (@VALUE '@FOCUSNODE' 'DOWN-DNAME') (@VALUE '@FOCUSNODE' 'DOWN-DTYPE')	
RULE NO.	L (条件部) H (件部) S (部)	R (実行部) H (部)
1	(EQUAL L-DEF 1) (EQUAL (@VALUE 'ND-ABEND-C7' 'HI-SEIKIKA') 'NO')	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7' ATTRIBUTES: HI-SEIKIKA = 'YES TAISAKU = 1 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME エラータイプ1で非正規化データ発見。結果出力KSを発行する。
2	(EQUAL L-DEF 3) (EQUAL (SEARCH-ABNORMAL L-DTYPE L-D-DTYPE) 'YES) 3) (EQUAL (@VALUE 'ND-ABEND-C7' 'HI-SEIKIKA') 'NO') 定義値タイプが3で、対象データと下位ルーチンでのデータが非正規化データの原因になっている。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7' ATTRIBUTES: HI-SEIKIKA = 'YES TAISAKU = 2 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME ERROR-DTYPE=L-DTYPE ERROR-D-RNAME=L-D-RNAME ERROR-D-DNAME=L-D-DNAME ERROR-D-DTYPE=L-D-DTYPE エラータイプ2で非正規化データ発見。結果出力KSを発行する。

3	<pre>(OR (EQUAL L-DEF 4) (EQUAL L-DEF 5)) (EQUAL (@VALUE @FOCUSNODE 'COMMON-ASK) 'NO) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'NO)</pre> <p>定義値タイプが4または、5であるとき、COMMON変数の先頭アドレスが一致していない。</p>	<p>(1) PROPOSE</p> <pre>CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : HI-SEIKIKA = 'YES TAISAKU = 3 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME ERROR-DTYPE=L-DTYPE ERROR-D-RNAME=L-D-RNAME ERROR-D-DNAME=L-D-DNAME ERROR-D-DTYPE=L-D-DTYPE</pre> <p>エラータイプ3で非正規化データ発見。結果出力KSを発行する。</p>
4	<pre>(EQUAL L-DEF 8) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'NO) (OR (EQUAL (@VALUE @FOCUSNODE 'DOWN-ASK) 1) (AND (EQUAL (@VALUE @FOCUSNODE 'DOWN-ASK) 3) (EQUAL (SEARCH-ABNORMAL L-DTYPE L-D-DTYPE) 'YES))) (@VALUE @FOCUSNODE 'DOWN-DEF-RNAME))</pre> <p>定義値タイプが8で、下位ルーチンでの定義値が1かまたは、定義値が3で非正規化データである。</p>	<p>(1) PROPOSE</p> <pre>CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : HI-SEIKIKA = 'YES TAISAKU = 4 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME ERROR-DTYPE=L-DTYPE ERROR-D-RNAME= (@VALUE @FOCUSNODE 'DOWN-DEF-RNAME)</pre> <p>エラータイプ4で非正規化データ発見。結果出力KSを発行する。</p>
5	<pre>(EQUAL L-DEF 9) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'NO)</pre> <p>定義値タイプが9である。</p>	<p>(1) PROPOSE</p> <pre>CHANGE TYPE: SUPERSEDE EVENT NAME: END-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : HI-SEIKIKA = 'YES TAISAKU = 5 ERROR-RNAME=L-RNAME ERROR-DNAME=L-DNAME ERROR-DTYPE=L-DTYPE ERROR-D-RNAME=L-D-RNAME ERROR-D-DNAME=L-D-DNAME ERROR-D-DTYPE=L-D-DTYPE</pre> <p>エラータイプ5で非正規化データ発見。結果出力KSを発行する。</p>
6	<pre>(EQUAL L-DEF 2) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'NO)</pre> <p>定義値タイプが2である。</p>	<p>(1) PROPOSE</p> <pre>CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : @FOCUSNODE ATTRIBUTES : BACK = 0 EQV-BACK = 2</pre> <p>バック・トラックタイプを2にする。</p>

7	<pre>(EQUAL L-DEF 3) (EQUAL (SEARCH-ABNORMAL L-DTYPE L-D-DTYPE) 'NO) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N O) 定義値タイプが3である。</pre>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : @FOCUSNODE ATTRIBUTES : BACK = 0 EQV-BACK = 3 バック・トラックタイプを3にする。</p>
8	<pre>(EQUAL L-DEF 8) (EQUAL (@VALUE @FOCUSNODE 'DOWN-ASK) 2) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N O) 定義値タイプが8であり、かつ下位ルーチンの定義値タイプが2である。</pre>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : @FOCUSNODE ATTRIBUTES : BACK = 0 EQV-BACK = 2 バック・トラックタイプを2にする。</p>
9	<pre>(EQUAL L-DEF 8) (EQUAL (@VALUE @FOCUSNODE 'DOWN-ASK) 3) (EQUAL (SEARCH-ABNORMAL L-DTYPE L-D-DTYPE) 'NO) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N O) 定義値タイプが8であり、かつ下位ルーチンの定義値タイプが3で非正規化データでない。</pre>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : @FOCUSNODE ATTRIBUTES : BACK = 0 EQV-BACK = 3 バック・トラックタイプを3にする。</p>
10	<pre>(NOT (@VALUE 'ND-ABEND-C7 'BACK-FLG)) (EQUAL L-DEF 4) (EQUAL (@VALUE @FOCUSNODE 'COMMON-ASK) 'Y ES) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N O) 定義値タイプが4であり、かつバック・トラック中でなく、COMMON変数の先頭アドレスが一致している</pre>	<p>(1) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : ABEND-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : QUEUE = (ASK-QUEUE-COMMON L-Q L-RNAME L-DNAME L-DTYPE 'NO) COMMON変数の間接定義値をセットする。</p>
11	<pre>(NOT (@VALUE 'ND-ABEND-C7 'BACK-FLG)) (EQUAL L-DEF 5) (EQUAL (@VALUE @FOCUSNODE 'COMMON-ASK) 'Y ES) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N O) 定義値タイプが5であり、かつバック・トラック中でなく、COMMON変数の先頭アドレスが一致している</pre>	<p>(1) EXECUTE EXPRESSION : (SETQ L-Q-WORK (ASK-QUEUE-COMMON L-Q L-RNAME L-DNAME L-DTYPE 'YES)) COMMON変数の直接定義値を作業変数にセットする。</p> <p>(2) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : ABEND-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : QUEUE = (CDR L-Q-WORK))さらに上位ルーチンを探索するためのKSを発行する。</p> <p>(3) PROPOSE CHANGE TYPE : SUPERSEDE EVENT NAME : DUMMY NODE : @FOCUSNODE ATTRIBUTES : EQV-K-DEF-VALUE = (CAR L-Q-WORK) 間接定義値タイプを保存する。</p>

12	<pre>(NOT (@VALUE 'ND-ABEND-C7 'BACK-FLG)) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) (EQUAL L-DEF 6) 定義値タイプが6であり、かつバック・トラック中でない。</pre>	<p>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: ABEND-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES: QUEUE = (ASK-QUEUE L-Q L-RNAME L-DNAME L-DTYPE 'NO) 対象データの実引数の定義値タイプをセットする。</p>
13	<pre>(NOT (@VALUE 'ND-ABEND-C7 'BACK-FLG)) (EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) (EQUAL L-DEF 7) 定義値タイプが7であり、かつバック・トラック中でない。</pre>	<p>(1) EXECUTE EXPRESSION: (SETQ L-Q-WORK (ASK-QUEUE L-Q L-RNAME L-DNAME L-DTYPE 'YES)) 対象データの実引数の定義値タイプを作業変数にセットする。</p> <p>(2) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: ABEND-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES : QUEUE = (CDR L-Q-WORK))さらに上位ルーチンを探索するためのKSを発行する。</p> <p>(3) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : @FOCUSNODE ATTRIBUTES : EQV-K-DBP-VALUE = (CAR L-Q-WORK) 間接定義値タイプを保存する。</p>
14	<pre>(EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) (@VALUE 'ND-ABEND-C7 'QUEUE) (@VALUE 'ND-ABEND-C7 'BACK-FLG) バック・トラック中で、かつまだ探索キューが空でない。</pre>	<p>(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: ABEND-C7 NODE : 'ND-ABEND-C7 ATTRIBUTES: NODE-NAME = @FOCUSNODE BACK-FLG = NIL</p>
15	<pre>(EQUAL (@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) 'N 0) (NOT (EQUAL (@VALUE @FOCUSNODE 'BACK) 0)) (NOT (EQUAL (@VALUES @FOCUSNODE 'DOWN-ROUTINE) 'ND-ABEND-C7)) バック・トラックタイプが指定(0以外)されていて、エラールーチンまでバック・トラックしていない。</pre>	<p>(1) PROPOSE CHANGE TYPE: NULL EVENT NAME: BACK-C7 NODE : @FOCUSNODE</p> <p>(2) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : 'ND-ABEND-C7 ATTRIBUTES : BACK-FLG=T</p>

KS name	KS-BACK-C7 (FORWARD)		C7バック・トラック KS					
trigger	BACK-C7							
strategy	MULTIPLE MANYTIMES							
LOCAL-VARIABLE	EXPRESSION							
L-D-DNAME L-K L-EK L-R L-D L-ED L-DT L-EDT L-D-D L-D-DT L-BACK L-EBACK L-NODE	$(@VALUE '@FOCUSNODE 'DOWN-DNAME)$ $(@VALUES '@FOCUSNODE 'DOWN-ROUTINE 'K-DEF-VALUE)$ $(@VALUES '@FOCUSNODE 'DOWN-ROUTINE 'EQV-K-DEF-VALUE)$ $(@VALUES '@FOCUSNODE 'DOWN-ROUTINE 'RNAME)$ $(@VALUES '@FOCUSNODE 'DOWN-ROUTINE 'DNAME)$ $(@VALUES '@FOCUSNODE 'DOWN-ROUTINE 'EQV-DNAME)$ $(@VALUES '@FOCUSNODE 'DOWN-ROUTINE 'DTYPE)$ $(@VALUES '@FOCUSNODE 'DOWN-ROUTINE 'EQV-DTYPE)$ $(@VALUES '@FOCUSNODE 'DOWN-ROUTINE 'DOWN-DNAME)$ $(@VALUES '@FOCUSNODE 'DOWN-ROUTINE 'DOWN-DTYPE)$ $(@VALUE '@FOCUSNODE 'BACK)$ $(@VALUE '@FOCUSNODE 'EQV-BACK)$ $(@VALUE '@FOCUSNODE 'DOWN-ROUTINE)$							
RULE NO.	L (条) H (件) S (部)	R (実) H (行) S (部)						
1	$(NOT (EQUAL L-BACK 0))$ $(EQUAL L-D-DNAME L-D)$ バック・トラックタイプが指定(0以外)され、バック・トラックの対象データが下位ルーチンの対象データと直接的に引数で結合されている。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : L-NODE ATTRIBUTES : DEF-VALUE = L-BACK バック・トラックして下位ルーチンに戻り定義値をセットする。						
2	$(NOT (EQUAL L-EBACK 0))$ $(EQUAL L-D-DNAME L-ED)$ バック・トラックタイプが指定(0以外)され、バック・トラックのEQUIVALENCEデータが下位ルーチンの対象データと直接的に引数で結合されている。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : L-NODE ATTRIBUTES : EQV-DEF-VALUE=L-EBACK バック・トラックして下位ルーチンに戻り定義値をセットする。						
3	$(NOT (EQUAL L-BACK 0))$ $(ASSOC L-D-DNAME L-K)$ バック・トラックタイプが指定(0以外)され、バック・トラックの対象データが下位ルーチンの対象データの間接定義変数と結合されている。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : L-NODE ATTRIBUTES : K-DEF-VALUE= $(K-CHANGE L-K L-D-DNAME L-BACK)$ バック・トラックして下位ルーチンに戻り間接定義値をセットする。						

4	(NOT (EQUAL L-D-DNAME L-K)) (ASSOC L-D-DNAME L-K) (K-DEF-FILL L-K) 対象データが下位ルーチンの対象データの間接定義変数と結合されている。間接定義値がすべてセットされた。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : L-NODE ATTRIBUTES : DEF-VALUE = (ASK-D L-R L-D L-K) すべての間接定義値から定義値を質問してセットする。
5	(NOT (EQUAL L-EBACK) 0) (ASSOC L-D-DNAME L-EK) バック・トラックタイプが指定(0以外)され、バック・トラックのEQUIVALENCEデータが下位ルーチンの対象データの間接定義変数と結合されている。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : L-NODE ATTRIBUTES : EQV-K-DEF-VALUE = (K-CHANGE L-EK L-D-DNAME L-EBACK) バック・トラックして下位ルーチンに戻り間接定義値をセットする。
6	(NOT (EQUAL L-EBACK) 0) (ASSOC L-D-DNAME L-EK) (K-DEF-FILL L-EK) バック・トラックタイプが指定(0以外)され、バック・トラックのEQUIVALENCEデータが下位ルーチンの対象データの間接定義変数と結合されている。 間接定義値がすべてセットされた。	(1) PROPOSE CHANGE TYPE: SUPERSEDE EVENT NAME: DUMMY NODE : L-NODE ATTRIBUTES : EQV-DEF-VALUE = (ASK-D L-R L-ED L-EK) すべての間接定義値から定義値を質問してセットする。
7	(NOT (EQUAL L-BACK) 0) バック・トラックタイプが指定(0以外)されている。	(1) PROPOSE CHANGE TYPE: NULL EVENT NAME: KENSAKU2-C7 NODE : L-NODE 下位ルーチンの元データが非正規化データかを調べるため元データ検索KSを発行する
8	(NOT (EQUAL L-BACK) 0) バック・トラックタイプが指定(0以外)されている。	(1) PROPOSE CHANGE TYPE: NULL EVENT NAME: KENSAKU1-C7 NODE : L-NODE 下位ルーチンのEQUIVALENCEデータが非正規化データかを調べるためEQUIVALENCEデータ検索KSを発行する。

KS name	KS-END-C7 (FORWARD)	C7結果出力KS
trigger	END-C7	
strategy	MULTIPLE MANYTIMES	
LOCAL-VARIABLES		EXPRESSION
L-ABNORMAL L-TAISAKU L-ERROR-RNAME L-ERROR-DNAME L-ERROR-DTYPE L-ERROR-D-RNAME L-ERROR-D-DNAME L-ERROR-D-DTYPE L-ERROR-EQV-DNAME L-ERROR-EQV-DTYPE		(@VALUE 'ND-ABEND-C7 'HI-SEIKIKA) (@VALUE 'ND-ABEND-C7 'TAISAKU) (@VALUE 'ND-ABEND-C7 'ERROR-RNAME) (@VALUE 'ND-ABEND-C7 'ERROR-DNAME) (@VALUE 'ND-ABEND-C7 'ERROR-DTYPE) (@VALUE 'ND-ABEND-C7 'ERROR-D-RNAME) (@VALUE 'ND-ABEND-C7 'ERROR-D-DNAME) (@VALUE 'ND-ABEND-C7 'ERROR-D-DTYPE) (@VALUE 'ND-ABEND-C7 'ERROR-EQV-DNAME) (@VALUE 'ND-ABEND-C7 'ERROR-EQV-DTYPE)
RULE NO.	L (条 件 部)	R (実 行 部)
1	(EQUAL L-ABNORMAL 'YES) (>= L-TAISAKU 1) (<= L-TAISAKU 6) 非正規化データが発見された。	(1) EXECUTE EXPRESSION : (PROGN (ERASE) (FG : DISPLAY ("異常終了の原因" 2 3 HXA) ("アベンドコードOC7 は、ベクトル演算 をするデータが," 4 3) ("非正規化データの場合に出力されます 。" 5 3) ("*****" 7 3 HB) (" * 診断結果 *" 8 3 HB) ("*****" 9 3 HB)) FORMAT "/N /N") FORMAT "/C /C /C /N" "ルーチン" L-ERROR-RNAME "において") COND ((EQUAL L-TAISAKU 1)) FORMAT "/C に文字型・HEX データが 代入されている。/N" L-ERROR-DNAME)) ((EQUAL L-TAISAKU 2)) (PROGN (FORMAT "データ/C が宣言タ イプ/C であるのに対し、下位ルーチン/ C における/N" L-ERROR-DNAME L-ERROR -DTYPE L-ERROR-D-RNAME) (FORMAT "引数およびCOMMON変数/C の 宣言タイプが/C であるためと考えられま す。/N /N" L-ERROR-DNAME L-ERROR-DTYPE L-ERROR-RNAME) (FORMAT "この宣言タイプが一致するよ うに修正してください。/N"))))

```

( (EQUAL L-TAISAKU 3 )
  (PROGN (FORMAT "/C を含んでいるCOMMON文中の変数のタイプ・配列の長さが/N"
    L-ERROR-DNAME)
    (FORMAT "/C /N /C /N" ルーチンによって一致していない可能性があります。" " 各ルーチンにあるCOMMON文を見直してください。" ) )
  ( (EQUAL L-TAISAKU 4 )
  (PROGN
    (FORMAT
      " 下位ルーチン/C との引数の宣言タイプ・配列の長さが一致/N" L-ERROR-D-RNAME
      (FORMAT " していられない可能性があります。/N" ) (FORMAT " 下位ルーチン/C 以下を調べてください。" L-ERROR-D-RNAME ) ) )
  ( (EQUAL L-TAISAKU 5 )
  (PROGN
    (FORMAT
      " /C が未定義であるためと考えられます。/N" L-ERROR-DNAME )
      (FORMAT " 未定義変数をなくすようにしてください。" ) )
  ( (EQUAL L-TAISAKU 6 )
  (FORMAT
    " /C と/C の宣言タイプ・配列の長さが一致していないためと考えられます。"
    L-ERROR-EQV-DNAME L-ERROR-DNAME ) ) )
  (TERPRI TERMINAL-OUTPUT)
  (WAIT " ENTER KEY を押してください。" )
)

```

(2) PROPOSE

```

CHANGE TYPE: SUPERSDE
EVENT NAME: INIT
NODE : 'ND-ABEND-C7
ATTRIBUTES :
FIRST-ND = 'YES
FIRST = 'YES
KENSAKU-FIRST = 'YES
HI-SEIKIKA = 'NO
制御変数を初期化する。

```

2	T	<pre> (1) EXECUTE EXPRESSION : (PROGN (ERASE) (FG : DISPLAY ("異常終了の原因" 2 3 HXA) ("アベンドコード0C7は、ベクトル演算 をするデータが、" 4 3) ("非正規化データの場合に出力されます 。" 5 3) ("*****" 7 3 HB) ("* 診断結果 *" 8 3 II B) ("*****" 9 3 II B) (FORMAT " /N /N ") (NOKS) (TERPRI TERMINAL-OUTPUT) (WAIT " ENTER KEY を押してください。 ")) </pre> <hr/> <pre> (2) PROPOSE CHANGE TYPE: SUPERSDE EVENT NAME: INIT NODE : 'ND-ABEND-C7 ATTRIBUTES : FIRST-ND = 'YES FIRST = 'YES KENSAKU-FIRST = 'YES II-SEIKIKA = 'NO 制御変数を初期化する。 </pre>
---	---	---