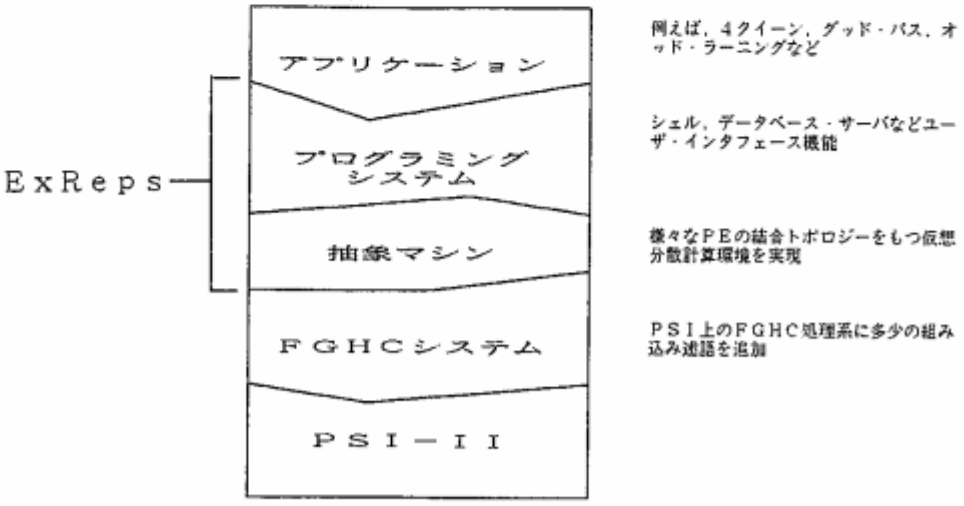
題　名	実験的リフレクティブ・プログラミング・システム　「ＥｘＲｅｐｓ」
目　的	核言語（ＧＨＣ）による，メタインタプリタの段階的拡張及びそれによる並列計算機の記述から，仮想並列ハードウエア上の環境を提供し，ＧＨＣのリフレクト機能拡張の研究開発を支援する．
概　要 及び 特　徴	〔概要〕　(1) メタインタプリタの段階的拡張 ・４行メタインタプリタ ・キューとリダクション・カウンタを持つメタインタプリタ ・変数管理を行うメタインタプリタ (2) 抽象マシンとプラグマ ・様々な抽象マシンの構築とプラグマによる指示 (3) プログラミング・システム上でのプログラム実行 ・４クイーン　・オッドラーニング　・グッドパス (4) リフレクション機能の実現 ・プロセス・キューの動的表示 ・リダクション・カウントの動的コントロール ・動的ロード・バランス 〔特徴〕　(1) 全てＧＨＣにより記述されている (2) 分散インプリメントに対応 (3) インタプリタベースの柔らかなシステム
構　成	

デ　　モ　　内　　容　（1／3）

1　GHCメタインタプリタの段階的拡張

GHCの最も簡単なメタインタプリタは4行で記述できるが，さらにこれを段階的に拡張することで，さまざまな機能を獲得することができる．以下に，各種メタインタプリタを示す．

①4行メタインタプリタ

```
exec(true):- true| true.
exec((P,Q)):- true | exec(P),exec(Q).
exec(Goal):- nonsys(Goal) |
    reduce(Goal,ReducedGs),
    exec(ReducedGs).
exec(Goal):- sys(Goal) |
    sys_exe(Goal).
```

②スケジューリング・キューとリダクション
カウンタを管理するメタインタプリタ

```
exec(G,Res):- true|
    exec([G|T],T,Res,0).

exec([Goal|H],T,Res,Rc):-
    nonsys(Goal),Goal\=true,Goal\=fail |
    reduce(Goal,H,T,NH,NT,Rc,NRc),
    exec(NH,NT,Res,NRc).
exec([Goal|H],T,Res,Rc):-
    sys(Goal) |
    sys_exe(Goal,T,NT,Rc,NRc),
    exec(H,NT,Res,NRc).
exec([true|H],T,Res,Rc):- true |
     exec(H,T,Res,Rc).
exec(T,T,Res,Rc):- true|
    Res=success(Rc).
exec([fail|H],T,Res,Rc):- true |
    Res=fail(Rc).
```

③変数管理をするメタインタプリタ

```
m_ghc(FGoal,Out):- true |
    transfer(FGoal,NGoal,1,Id,Env),
    exec([NGoal|T],T,Id,Env,NEnv,Res),
    print_result(Res,NGoal,Out,NEnv).

exec([G|H],T,Id,Env,NEnv,Res):-
    nonsys(G),G\=true,G\=fail|
    reduce(G,H,T,NH,NT,Id,NId,Env,Env1),
    exec(NH,NT,NId,Env1,NEnv,Res).
exec([G|H],T,Id,Env,NEnv,Res):-
    sys(G) |
    sys_exe(G,T,NT,Env,Env1),
    exec(H,NT,Id,Env1,NEnv,Res).
exec(T,T,Id,Env,NEnv,Res):- true|
    Res=success,NEnv=Env.
exec([true|H],T,Id,Env,NEnv,Res):- true |
    exec(H,T,Id,Env,NEnv,Res).
exec([fail|H],T,Id,Env,NEnv,Res):- true |
    Res=fail,NEnv=Env.
```

また，図1にこれらメタインタプリタ上での例題実行の一部を示す．

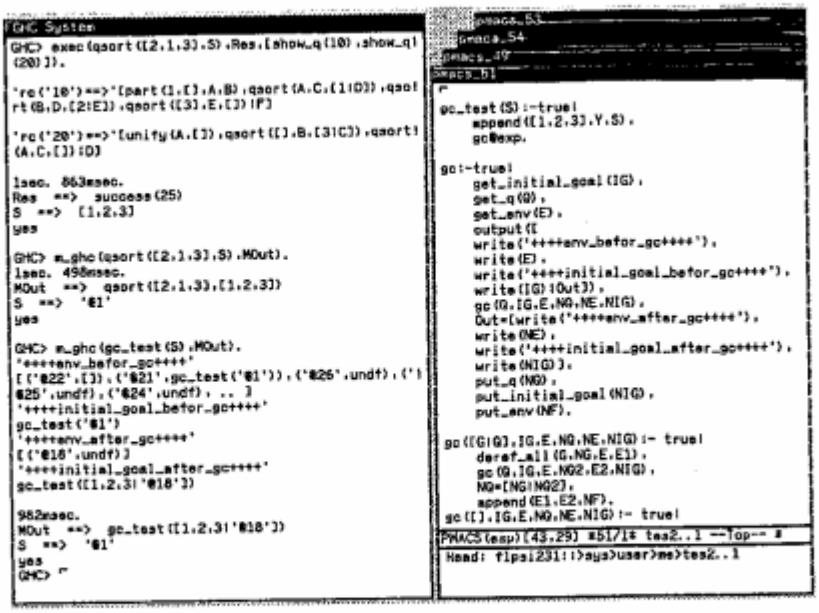


図1　メタインタプリタ上での例題実行

2　抽象ＧＨＣマシンとプラグマ

抽象ＧＨＣマシンは，複数のＰＥがつながれたネットワークから成る．この上で実行されるゴールは，プラグマを付けることによりネットワークを通ってＰＥ間を渡り歩くことができる．図２に，本システムで実現したネットワークの形を示す．

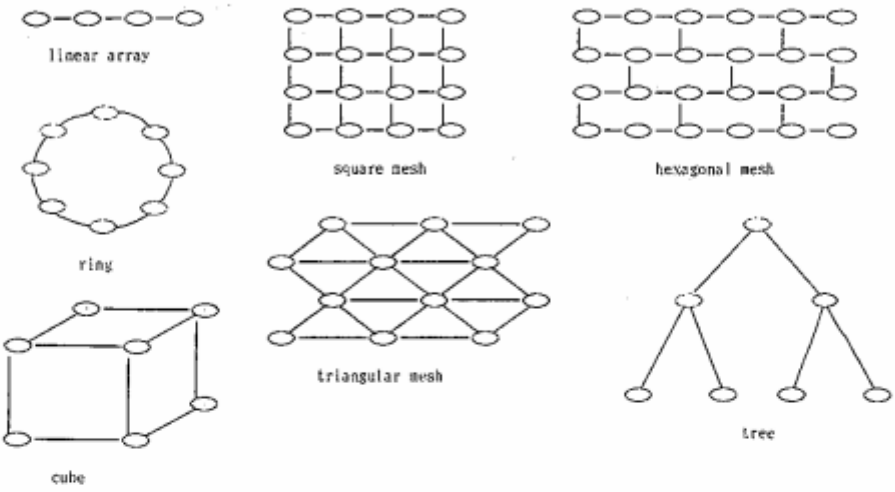


図２　本システムで実現した抽象ＧＨＣマシンのネットワーク

３．プログラミング・システム上でのプログラム実行

構築した抽象ＧＨＣマシンの上にプログラミング・システムを立ち上げることができる．アプリケーション・プログラムはこのプログラミング・システム上で実行される．

図３では，４クイーンのプログラムを，抽象ＧＨＣマシン上で分散して実行した．図４では，グッドパスのプログラムの並列実行を視覚的に表示した．

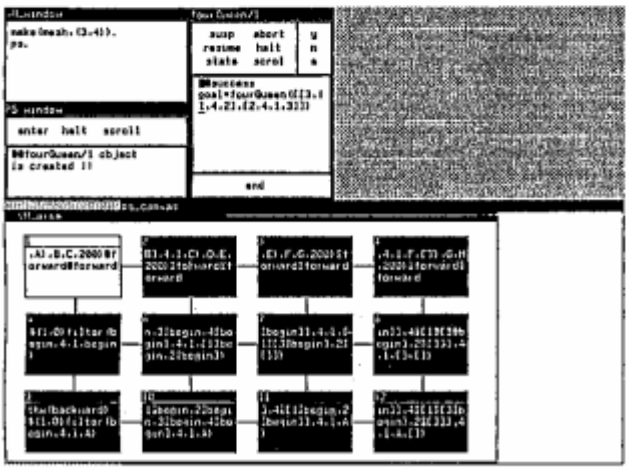


図３　４クイーン

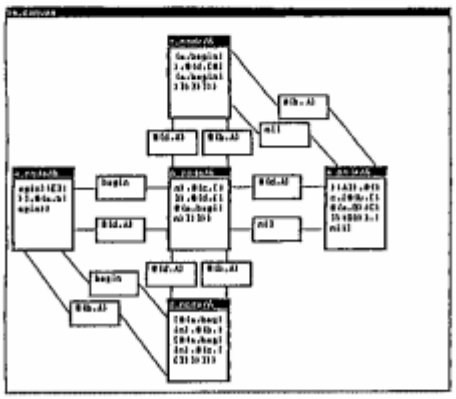


図４　グッドパス

4　リフレクト機能

リフレクト機能とは，プログラムを実行しながら，自分の現在の実行状態を取り出したり，変更したりする機能である．
リフレクト機能は，その言語による自己記述つまりメタインタプリタにより実現可能．

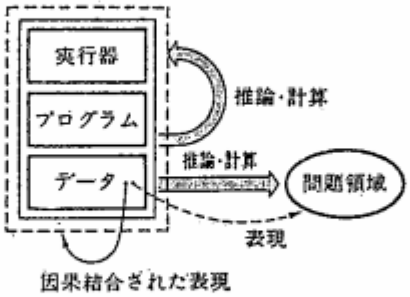


図5　リフレクテイブ・システム

本システムでは，拡張したメタインタプリタを用い，次に実行するリフレクト機能を実現した．
図6に，ユーザ・プロセスのキューを表示するプログラムをユーザが記述して実行した例を示す．図7に，ユーザ・プロセスに許されたリダクション数を動的に変更できるプログラムを実行した例を示す．

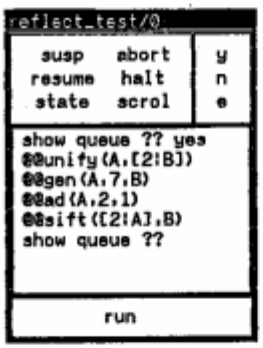


図6　プロセス・キューの動的表示

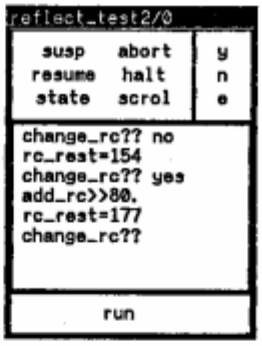


図7　リダクション・カウントの動的コントロール

図8に，マシン上で直接実行することによって，マシンの負荷分散が行えるプログラムを実行した例を示す．

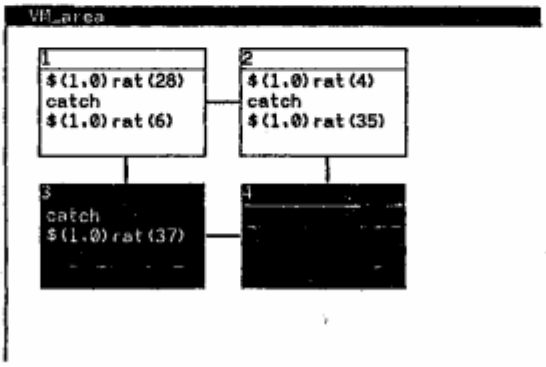


図8　動的ロード・バランス