並列自動証明システム *MGTP*

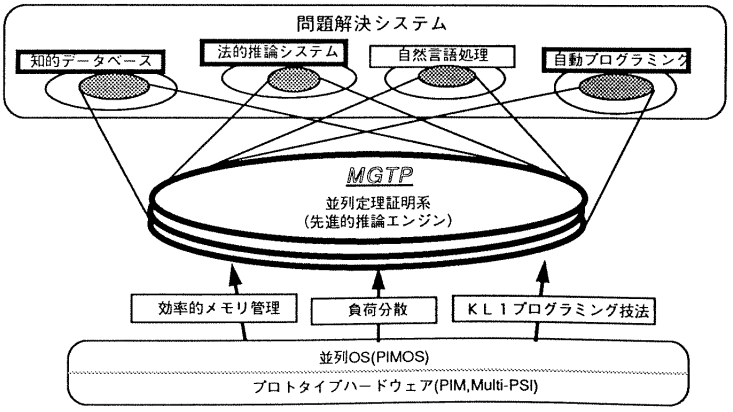
概要

本研究では並列推論マシン PIM/m 上の並列推論システムの開発を行なっている。現在 ICOT では、以下の 2 点を達成することを目標として、モデル生成法に基づいた定理証明系 MGTP の研究開発を進めている。

1. ロジックプログラミングと自動推論技術の結合及び並列推論マシンにおける並列処理技術の開発により、1 階述語論理の高速定理証明系を実現する。
2. 知的データベース、仮説推論システム、自然言語処理や自動プログラミングなどの分野に応用可能な先進的な推論エンジンを提供する。

特徴

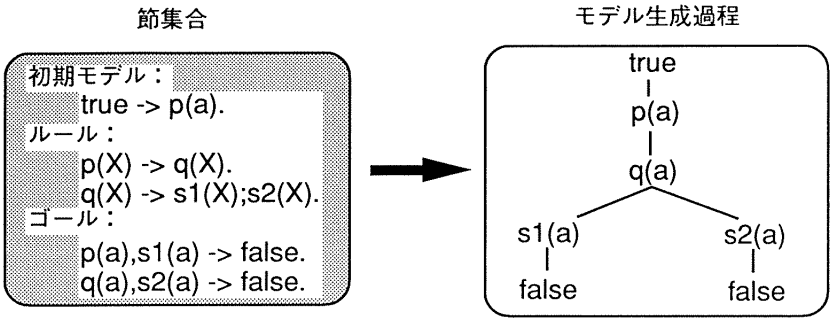
- KL1 の節コンパイル技術の応用とメタプログラミング機能の開発により、効率的な定理証明系を実現
- 高スケーラビリティを実現する AND/OR 並列処理技術の開発により、PIM/m(256 台) 上において線形台数効果を達成
- 定理証明系の開発実験環境を構築
- MGTP の応用研究 (仕様記述, アブダクション, 自動プログラミング)



FGCSにおけるMGTPの位置づけ

モデル生成法:

与えられた節集合から前向きに単位節（モデル）を生成する



・単位節が変数を含む問題 → 非基底問題

・単位節が変数を含まない問題 → 基底問題

問題例：

(a) Party 問題 - 非ホーン基底問題

例

6人の人間で、あるパーティを開いたとき、お互い顔見知りかまたは全く知らないかのどちらかである3人のグループを作ることができる。

```
true-->person(1),person(2),person(3),person(4),
       person(5),person(6).

person(X),person(Y),X>Y-->f(X,Y);nf(X,Y).

f(X1,X2),f(X2,X3),f(X3,X1)-->false.
nf(X1,X2),nf(X2,X3),nf(X3,X1)-->false.
```

(b) Condense Detachment 問題

- ホーン非基底問題
- 群論、環論、含意論理

例

```
p(X),p(e(X,Y))-->p(Y).
      p(e(e(e(a,e(b,c)),c),e(b,a)))-->false.
      true-->p(e(A,e(e(B,e(C,A)),e(C,B)))).
```

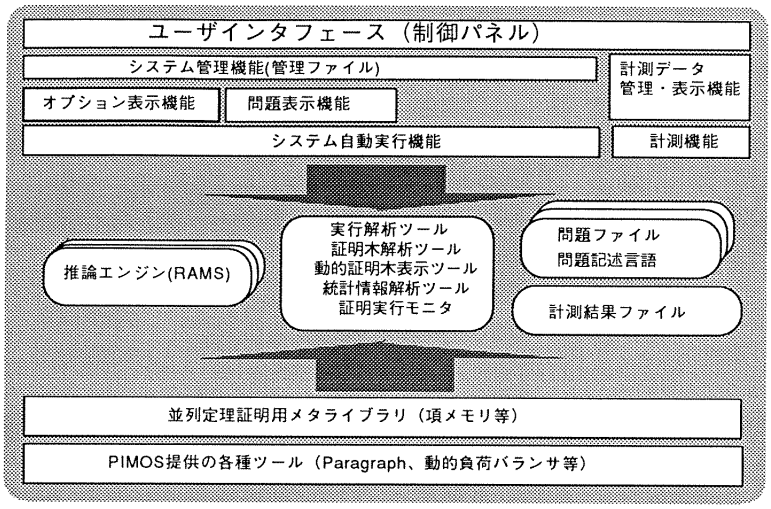
二つのMGTP:

	Ground MGTP	Non-Ground MGTP
対象問題	非ホーン基底問題	ホーン非基底問題
応用	例：データベース	例：数学の定理
プログラミング技術	・KL1変数を直接利用 ・問題節をKL1節に変換 ・KL1のヘッドユニフィケーションを用いて効率化	・基底項による変数表現 ・問題節を解釈する ・メタライブラリを利用 例：オカーチェック付きユニフィケーション
並列化	AND／OR 並列	AND 並列

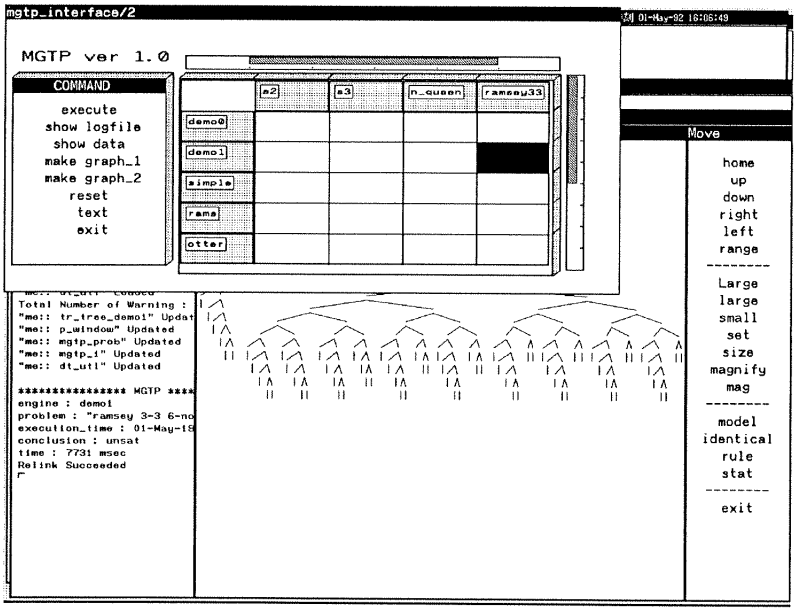
MGTPの高速化技術:

高速化に対する課題	解決方法（高速化技術）
連言照合の冗長性	RAMS(RamifiedStack) MERC(Multi-Entry Repeated Combination)
不適切なルールの適用	False節優先評価
冗長モデル生成	遅延モデル生成
並列化	非ホーン問題に対するOR並列化 ホーン問題に対するAND並列化 （モデル分散／共有方式）
ユニフィケーション／包摂テストの負荷	節コンパイル技術 項メモリ

MGTP総合開発支援環境



システム構成



制御パネルと証明木解析ツール

MGTP の並列化

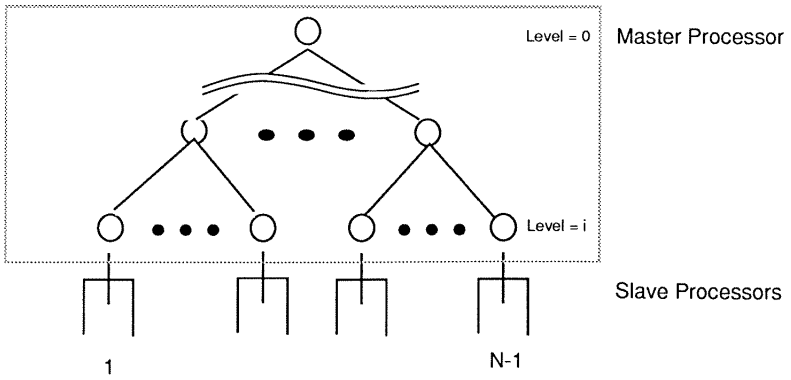
MGTP による証明過程では以下のプロセスに並列性が存在している。

- 場合分け
- 節の前件部の連言照合
- 包摂テスト

これらのプロセスに関して、場合分けにおける OR 並列性と節前件部の連言照合や包摂テストにおける AND 並列性に着目し、MGTP の並列化を行なった。

OR 並列化

非ホーン基底問題に対しては、場合分けによる OR 並列化が有効であることから、MERC 方式に基づいた OR 並列 MGTP を開発した。プロセス割り付け方式は、証明過程の際に発生する OR 並列のプロセスフォークを、プログラム資源の要件を満たすように制限した、‘制限付き OR’ 並列化に基づいたものである。この並列化を実現するため、深さ制限割り当て方式と呼ばれるプロセス割り付け方式を開発した。この方式では、マスタプロセッサが利用可能なスレーブプロセッサ数を越えるまでモデル候補を生成し、ある探索木のレベルで越えた時点で、スレーブプロセッサに残りのタスク (探索木の枝) を割り当てる。各スレーブプロセッサは割り当てられた探索木の枝の探索だけを行ない、他のプロセッサにはタスクを投げない。この方式では、プロセッサ間の通信が非常に小さいので、負荷分散が極めて有効に働く。



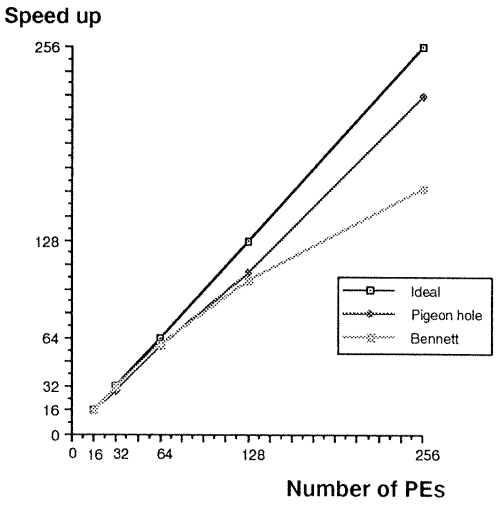
深さ制限付き割付け方式

Pigeon hole 問題

$true \rightarrow pigeon(1), pigeon(2), pigeon(3), pigeon(4), pigeon(5),$
$pigeon(6), pigeon(7), pigeon(8), pigeon(9), pigeon(10), pigeon(11).$
$pigeon(M) \rightarrow hole(M,1); hole(M,2); hole(M,3); hole(M,4); hole(M,5);$
$hole(M,6); hole(M,7); hole(M,8); hole(M,9); hole(M,10).$
$hole(M1,N), hole(M2,N), M1 \neq M2 \rightarrow false.$

Bennett's quasigroups

$true \rightarrow dom(1), dom(2), dom(3), dom(4), dom(5),$
$dom(6), dom(7), dom(8), dom(9), dom(10), dom(11).$
$dom(M), dom(N) \rightarrow$
$p(M,N,1); p(M,N,2); p(M,N,3); p(M,N,4); p(M,N,5); p(M,N,6);$
$p(M,N,7); p(M,N,8); p(M,N,9); p(M,N,10); p(M,N,11).$
$p(X,11,Y), Y+1 < X \rightarrow false.$
$p(E,X,Y), p(Y,E,Z), p(Z,E,U), X \neq U \rightarrow false.$
$p(X,X,U), X \neq U \rightarrow false.$
$p(X,Y,U), p(X,Y1,U), Y \neq Y1 \rightarrow false.$
$p(X,Y,U), p(X1,Y,U), X \neq X1 \rightarrow false.$



PIM/m 上の MGTP/G

AND 並列化

連言照合と包摂テストの並列化によって、ホーン問題に対する AND 並列化を行い、遅延モデル生成に基づいて MGTP の並列化を実現した。本システムでは、PE 台数によって証明が変わらない証明不変方式、(分散メモリアーキテクチャ上で各 PE が全モデルをコピーして持つ) モデル共有方式、マスター・スレーブ方式を採用した。

証明不変 並列定理証明器を開発する際の我々の方針は、得られた速度向上が並列化によるものなのか、戦略による探索空間の刈り込みによるものなのか、を明確に区別することである。証明変動方式において PE 台数を変えることは戦略を変えることになり、それによって超線形台数効果が得られることもあれば、逆に証明が長びいたりすることもある。これに対し証明不変方式では台数によって証明が変わらないので、投入した PE 台数に見合った速度向上を得ることが可能である。

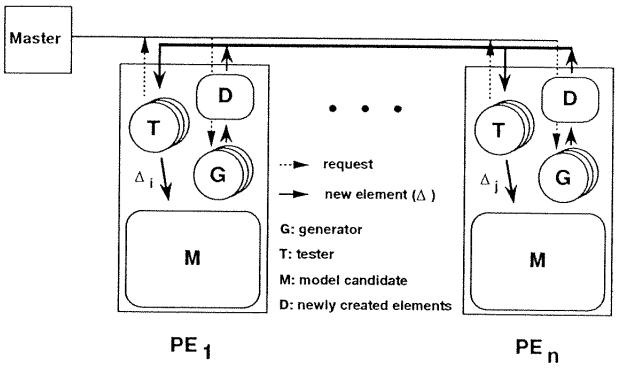
モデル共有 モデル共有方式の利点は、連言照合や包摂テストといった最もコストのかかる計算を他の PE と独立に最小の PE 間通信で行えることである。

マスター・スレーブ マスター・スレーブ方式では、逐次版 MGTP をスレーブ PE 上に置き、単にそれらとマスタ PE とつなぐことによって簡単に並列システムを構築することができる。スレーブプロセスは、暇になるとマスタプロセスからタスクを受けとれ、また各タスクの粒度はほぼ均等なので、良い負荷分散が得られる。

本システムでは、生成プロセスと包摂プロセスは要求駆動方式、棄却プロセスはデータ駆動方式で走行する。

本システムにおいて、並列化の際の最も大きな阻害要因は、包摂テストの逐次性である。この逐次性は、KL1 の同期機構によって最小に抑えることができる。また、KL1 のストリームを利用して、要求駆動制御が容易にかつ効率良く実現することができる。

要求駆動制御により、不必要なモデル拡張や包摂テストを抑制でき、さらに高い稼働率を得ることができる。高稼働率の達成は線形台数効果を得るためには必須である。



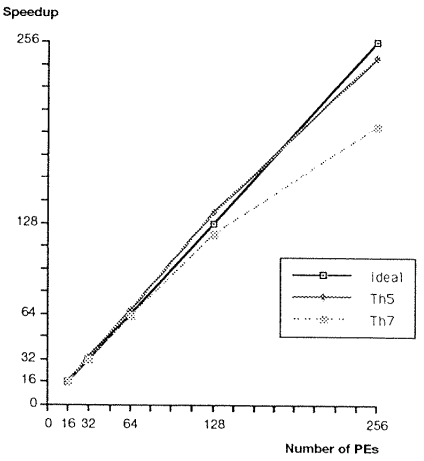
MGTP/N の負荷分散

定理 5

$true \rightarrow p(i(i(i(X,Y),Z),i(i(Z,X),i(U,X)))).$
$p(X), p(i(X,Y)) \rightarrow p(Y).$
$p(i(i(a,b),i(i(b,c),i(a,c)))) \rightarrow false.$

定理 7

$true \rightarrow p(i(X,i(Y,X))).$
$true \rightarrow p(i(i(n(X),n(Y)),i(Y,X))).$
$true \rightarrow p(i(i(X,Y),i(i(Y,Z),i(X,Z)))).$
$true \rightarrow p(i(i(i(X,Y),Y),i(i(Y,X),X))).$
$p(X), p(i(X,Y)) \rightarrow p(Y).$
$p(i(i(a,b),i(n(b),b(a)))) \rightarrow false.$



PIM/m 上の MGTP/N

MGTP 応用システム (1)
アブダクション・システム

概要

MGTP 上に開発したいくつかのアブダクション・システムのうち、図 1に示す構成によって表されるような、MGTP を用いたアブダクション・システムを実現するための 2 種類の構築方法 (MGTP+MGTP 方式、Skip 方式) を示す。与えられた入力はそれぞれのシステムに対応するように変換される。この 2 つのシステムについて、論理回路設計問題を例題として、デモンストレーションを行う。

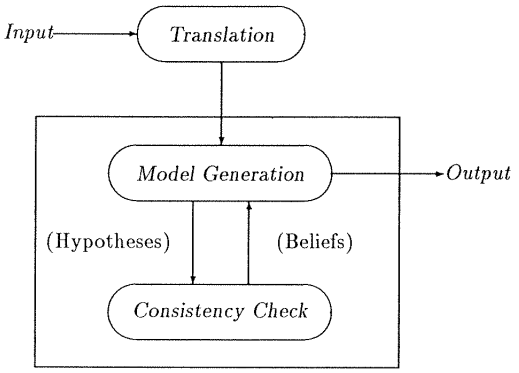


図 1: MGTP を用いたアブダクション・システム

アブダクション

ここではアブダクションを、論理式 (ホーン節) の集合 Σ、アトムの集合 Γ、閉論理式 G が与えられたときに、(Σ, Γ) からの G の説明を計算することと定義する。ここで、Γ の要素の基礎例の集合 E が (Σ, Γ) からの G の説明であるとは、(1) $\Sigma \cup E \models G$、(2) $\Sigma \cup E$ は無矛盾、を満たすこととする。

MGTP+MGTP 方式

このシステムでは、Γ の各要素 H は $fact(H, \{H\})$ のように変換され、さらに、Σ 中の各ホーン節

$$A_1 \wedge \ldots \wedge A_n \supset C$$

は

$$fact(A_1, E_1), \ldots, fact(A_n, E_n) \rightarrow fact(C, \mathrm{cc}(\bigcup_{i=1}^{n} E_i))$$

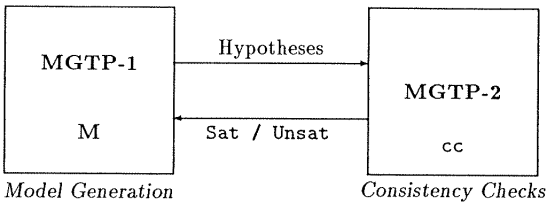


図 2: MGTP+MGTP

のような MGTP ルールに変換される。ここに、 E_i は Γ の基礎例の部分集合であり、アトム A_i が依存する仮説集合を表す。また、 cc は次のように定義される関数である。

$$\mathrm{cc}(E) = \begin{cases} E & \Sigma \cup E \text{ が無矛盾;} \\ \mathsf{nil} & \text{その他.} \end{cases}$$

MGTP-1 が新しい基礎アトムを導くたびに、組み合わされる仮説集合の無矛盾性が MGTP-2 によって検査される。この構成 (図 2) においては、一度に生成される複数の MGTP-2 に並列性がある。

Skip 方式

Σ の任意の節が、 その前件部に仮定可能なアトム $H_1, \ldots, H_m$ ($H_i \in \Gamma$, $m \geq 0$) を含んでいるとき、 すなわち、

$$A_1 \wedge \ldots \wedge A_l \wedge \underbrace{H_1 \wedge \ldots \wedge H_m}_{abducibles} \supset C$$

であるとき、 これを以下のような形式の MGTP ルールに変換して扱う:

$$A_1, \ldots, A_l \rightarrow H_1, \ldots, H_m, C \mid \neg \mathsf{K} H_1 \mid \ldots \mid \neg \mathsf{K} H_m \,.$$

つまり、 前件部における各仮説はマッチングの対象から除外され、右辺に *Skip* される。この方式では、 もしモデル候補が H と $\neg \mathsf{K} H$ とを含んでいたならば、次のスキーマによって棄却される:

$$\neg \mathsf{K} H, H \rightarrow \qquad \text{for every hypothesis } H \,.$$

このシステムでは、 MGTP が提供する OR 並列を利用した並列化がなされている。また、 この方式において、構成されるモデル候補の組合せ数を減少させるような知識を自動的に生成する方法についても示す。

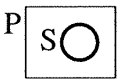
MGTP 応用 (2)
プロトコル仕様記述システム

概要

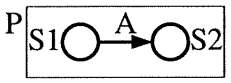
複雑なプロトコルの仕様を容易に記述できるように、プロトコル仕様記述言語 Ack を提案した。また、 その処理系を並列定理証明技術を利用して開発し、電話交換機のサービス記述に適用した。

Ack の記法

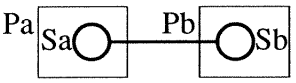
- 原子論理式



: プロセス P の状態が S である

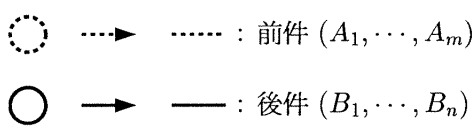


: プロセス P の状態がアクション A により S1 から S2 へ遷移する



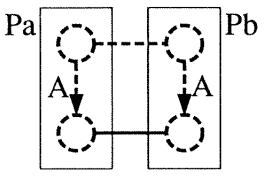
: 同時に、プロセス Pa の状態が Sa であり、プロセス Pb の状態が Sb である

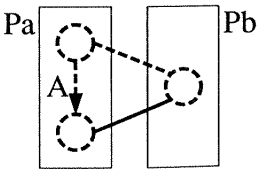
- 節 $(A_1 \wedge \cdots \wedge A_m \rightarrow B_1 \wedge \cdots \wedge B_n)$



Ack の公理

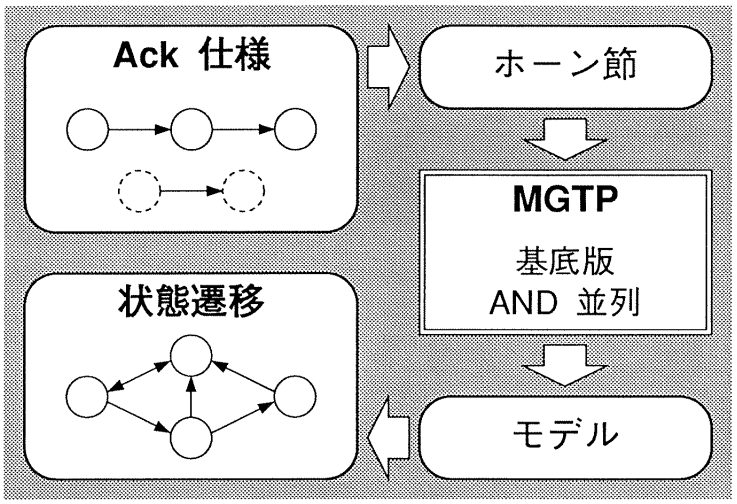
- 同期





ここで A $\notin$ Action(Pb)

状態遷移の合成



記述と合成の例