GHCのメッセージ指向の処理方式

ICOT
第2研究室
(株)三菱総合研究所

森　田　正　雄

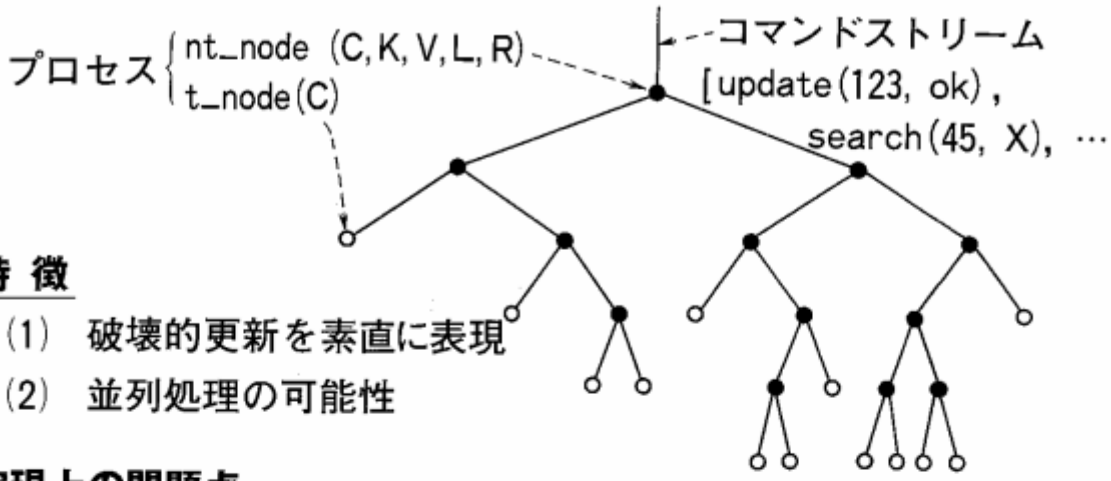
研究の動機

並行プロセスの性質	応用分野
計算指向，高並列性	データ駆動並列計算
	要求駆動計算
	動的データ構造
記憶指向，低並列性	データベース

- 従来の処理系は計算指向のプログラムに照準を当てて最適化
- 記憶指向のプログラムの最適化技法の開発は，並列論理型言語の応用分野の拡大に寄与

記憶指向のプログラムの性質

例：プロセスとストリームによる二分木の実現



特 徴

(1) 破壊的更新を素直に表現

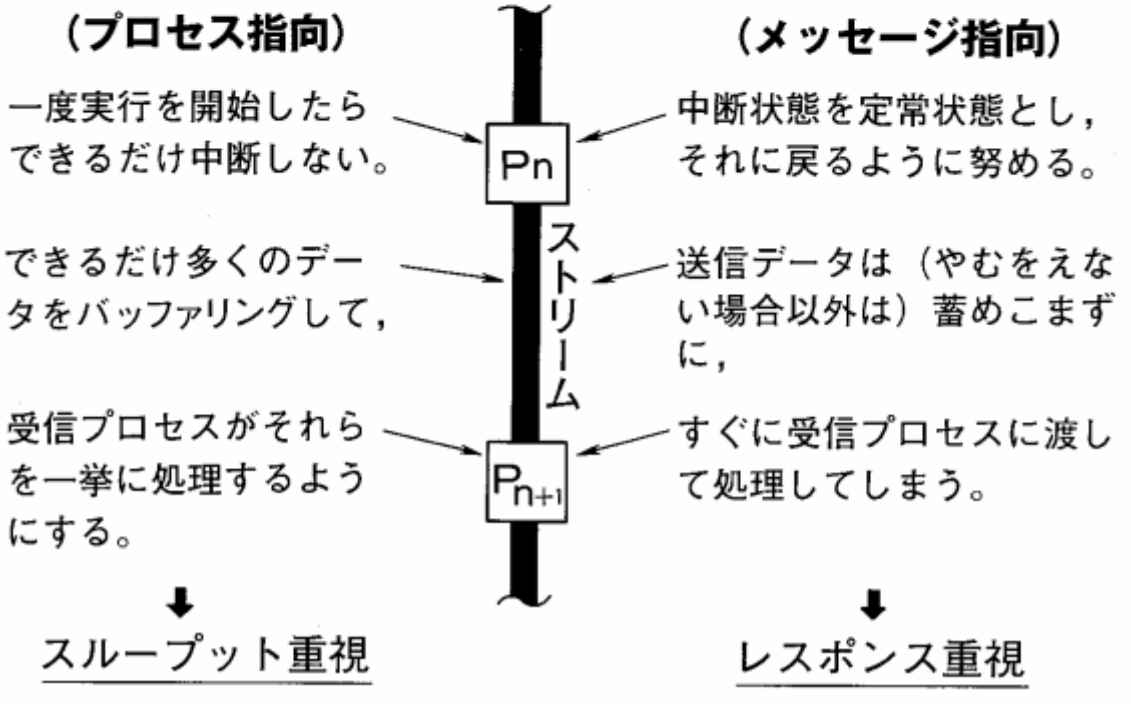
(2) 並列処理の可能性

実現上の問題点

(1) 頻繁に発生するプロセス切替えの時間効率

(2) ストリーム通信の空間効率

プロセス指向 vs. メッセージ指向スケジューリング



プロセス指向 vs. メッセージ指向スケジューリング

```
p([A | X'], Y) :- true | Y=[A | Y'], p(X', Y').
```

(データの受信)　　データAの送信　　次データの受信準備

ボディ・ゴールの実行順序

- プロセス指向：まず単一化ゴールを実行してデータA をバッファリングし，次に再帰呼出しを(TROによって) 効率よく実行する。
- メッセージ指向：まず再帰呼出しを実行して中断状態を回復し，次に単一化ゴールをメッセージ送信として効率よく実行する (データA と制御の両方をA の受信プロセスに渡してしまう)。

メッセージ送受信の解析

単一化をメッセージ送信にコンパイルするためには，静的プログラム解析が必須。

(1) メッセージ指向スケジューリングの効果が大きいのは，ストリーム (リスト) を用いた通信。
→ 型推論によるストリーム通信と非ストリーム通信の区別が必要。

(2) メッセージの送信時には，受信プロセスが存在し，メッセージを処理できる状態にしなければならない。
→ 受信プロセスを送信プロセスより先に実行する。
→ モード推論によるプロセス間通信の向きの解析が必要。

▶制約概念に基づくモード解析・型解析技法を開発した。

GHCプログラムのためのモード体系

設計目標

- プログラムの実行時に起きる通信の向きを静的に推論する。
- 単一化による通信のもたらす柔軟性を確保する。
  例：双方向通信（未完成メッセージ），ストリームのストリーム

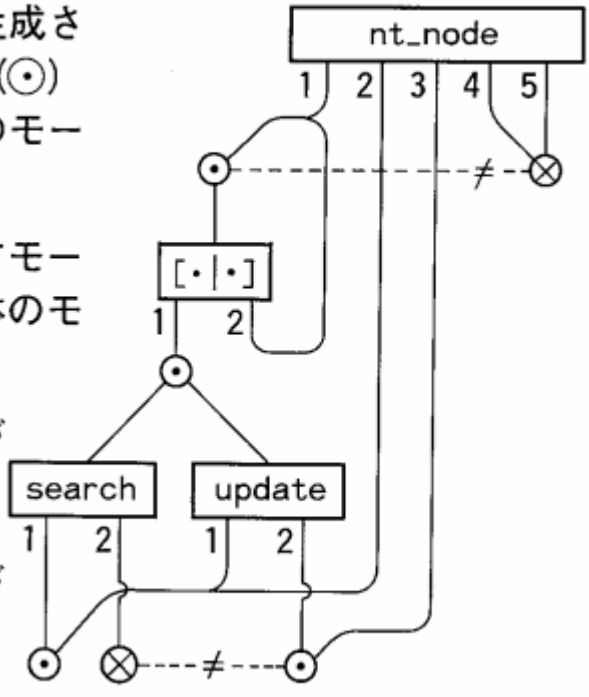
プログラムに対する制限

- 各共有変数を具体化できる出現はただひとつ(具体化は協調的)。
  - ボディの単一化は(occur check以外の理由では)失敗しない。
- Overloadした（複数のモードをもつ）述語は単一化以外にはない。

▶ モード体系の導入は，(Flat)GHCの仕様縮小の一方向でもある。

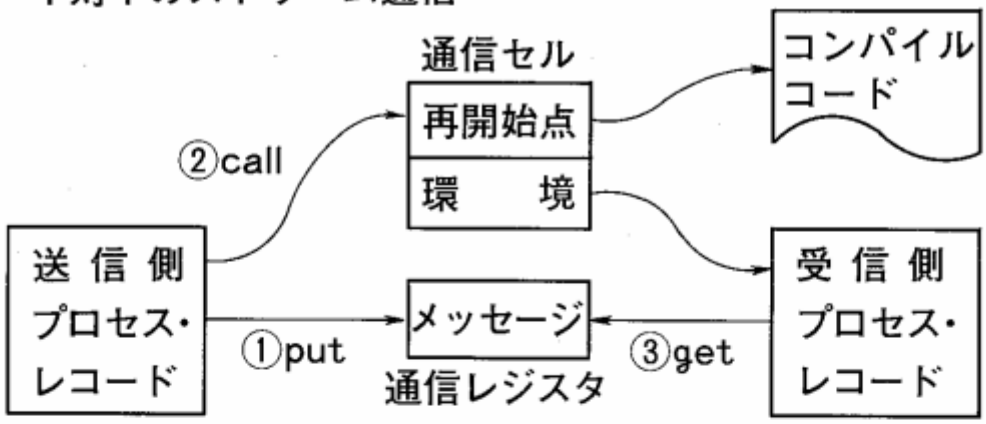
モード解析

- プログラムの実行によって生成されうる構造の各部に，入力（⊙）または出力（⊗）いずれかのモードを割り当てる。
- 各プログラム節が個別に課すモード制約を集めることで，全体のモードが定まる。
  - 大域的データフロー解析が不要。
  - モード宣言，推論，検査が統一的枠組で扱える。



メッセージ送受信の実現

(1)　1対1のストリーム通信



- ストリームは，リストでなく通信セルとして実現している。

(2)　1対1でないストリーム通信をどうするか？

一般のストリーム通信

通信形態	プログラム例	処理方式
(1) ⊗⟶⊙	:-p(X), c(X).	(基本形)
(2) ⊗⟶	:-p(X). c([A | X]) :-true | true.	ダミーの受信器の生成
(3) ⊗⟶⊙ ⊗⟶⊙	:-p(X), c1(X), c2(X). c(X) :-true | c1(X), c2(X).	メッセージ分配器の生成
(4) ⊗⟶⊙ ⊗⟶⊙ (非選択的)	非決定的ストリーム併合	基本形に同じ
(5) ⊗⟶⊙ ⊗⟶⊙ (選択的)	大小関係を保存した併合 ストリームの連結	即座に処理できないメッセージは受信側でバッファリング

ネイティブ・コードを用いた性能評価(VAX11/780上)

(1) プロセスの二分木（721項目）の検索（800件）

プロセス指向	1.04秒（一括検索），2.09秒(間欠的検索)	
メッセージ指向	0.75秒	
Cプログラム	0.31秒	

(2) 素数生成（2～1000)

プロセス指向	1.23秒	(データ駆動),	4.96秒	(要求駆動)
メッセージ指向	0.83秒		1.38秒	

(3) リストのnaive reverse

プロセス指向	53KRPS(最適化後)，	消費メモリO(n²)
メッセージ指向	55KRPS,	消費メモリO(n)

ま と め

(1) GHCのメッセージ指向の処理方式を提案し，詳細化した。
- 記憶指向のプログラムや要求駆動プログラムの効率を改善した。
- 一部の計算指向のプログラムの効率も改善した。
- 単一化をコルーチン間のデータと制御の移動に近い形にコンパイルすることに成功した。
  → バッファリングは本質的に必要な場合以外は行なわない。

(2) 制約概念に基づくモード解析技法を開発した。これによってはじめて上記の処理方式が現実のものとなった。

(3) 今後の課題は，並列処理系への本技法の適用と，実用的な処理系の開発である。