



論理的に同一であることが要求される。即ち、ゴールから起動される両プロセスで  $n(n=1, 2, \dots)$  番目に参照される要素は同一でなければならない。このため、このようなストリーム共有を行う場合、SDのコピーを生成し、両プロセスに渡す制御を行う。この処理を行うプリミティブを用いると、与えられた筋は以下のように書きなおせる。

$$p(X) :- \text{true} \mid \text{share\_stream}(X, X1, X2) \ \& \ q(X1), \\ r(X2).$$

変数XにSDが具体化されているとき、share\_stream(X, X1, X2) はXで示されるSDを新たにコピーし、それぞれ X1及び X2に具体化する。

しかし、これは次のような問題を生む。ストリームを含む構造データが変数Xに具体化されてプロセス間で共有される場合、ストリームのSDのコピーを生成するためには、構造データをコピーしなければならない。例えば、先に述べた節で、変数XがストリームSを含む構造体データ

$f(\dots, S, \dots)$ に具体化しているとき share\_stream は図2のようにSDの二つのコピー、S1及びS2を生成して  $f(\dots, S1, \dots)$ 及び  $f(\dots, S2, \dots)$ を作成し、述語q及びrの引数として渡す処理を行う必要がある。このようなコピーを share\_stream 実行時に eagerに生成するのはオーバーヘッドが大きい。即ち、分散環境における構造データコピー生成の問題を併用し、また、コピーしようとする構造データ中に一つでも未定義変数が存在すれば、後に変数がストリームに具体化されるかもしれないために具体化まで share\_stream 実行を中断しなければならない。

この回避策の一つとして上述のようなストリームを含む構造データの共有を許さない方式がある。即ち、Packed Streamを構造の要素としてユニファイできないようにすることである。しかし、この方式はあまりに制約が大きい。

もう一つは、変数に共有フラグを用意し、プロセス間で共有される変数とそうでない変数を区別する方法である。これは、前述の share\_stream に、Xが未定義変数であれば、その共有フラグをセットして共有変数に変えて、X1及び X2とユニファイする機能を持たせる。これによって、ゴールq及びrはその引数として共有変数を受取る。同様に、構造データである場合はその共有フラグをオンにし、共有構造データとする。なお、Xがアトム定数の場合はX1及び X2にそのまま定数がユニファイされる。このような共有フラグはユニフィケーション時にその部分構造に継承される。即ち、構造データを分解するユニフィケーションを行うとき、その要素として現われる変数や構造データの共有フラグはオンに設定され、それぞれ、共有変数及び共有構造データに更新される。SDのコピーは共有変数とSDとのユニフィケーションが行われたとき始めて行われる。このような方式をSDの遅延コピー方式と呼ぶ。

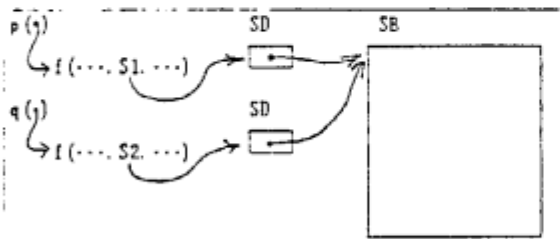


図2. 構造体データの eager なコピー

#### 4. 評価

上述のようなストリーム操作プリミティブをデータフローマシンの機械語として実現し、ソフトウェアシミュレーションを行った。本シミュレーションにおいては、Packed StreamのSDやSBを処理要素PE (Processing Element)の局所メモリLM (Local Memory)に格納するモデルを採用した。

シミュレーション対象プログラムとして 7queensを選び、表1のように、①最適化しない版、②TROによる最適化版、及び③TROと Packed Stream による最適化版の三つのコードについて評価した。

ここで、TRO (Tail Recursive Optimization)とは、末尾再帰呼出しを手続き呼出しとしてではなくループとして実現することによる最適化である。表1にこれらの結果を示す(但し、同表はPE 1台のときの結果である)。

表1. 実行サイクル数と実行命令数

	①	②	③
サイクル数	4835015	1887866	1543482
命令総数	625597	285807	258958

TROを行うことによって、手続き呼出しのオーバーヘッドが大幅に減少し、性能が約2.5倍向上した。さらに、Packed Streamを導入することによる最適化を行うと性能は約10%向上する。

#### 5. 終りに

GHCにおけるストリーム処理を効率化するために Packed Streamを導入し、その基本操作プリミティブを示した。シミュレーションによると、これを導入することにより性能向上が確かめられた。

現在、この操作プリミティブの実験機<sup>[1]</sup>上のインプリメンテーションを進めており、実験機による評価も行う予定である。この Packed Stream 導入による効果は命令の間の独立性が保証されているデータフローマシンよりも、むしろ現在、検討を進めている中期並列推論マシン<sup>[2]</sup>で大きいと考えられ、このマシン上での実現も検討している。

最後に、日頃御指導をいただくICOT内田室長、後藤研究員、及び、御討論いただくPIMグループ関係各位に深謝する。

[1] Ueda, K., "Guarded Horn Clauses," ICOT TR-103, June, 1985

[2] 後藤、他、「並列推論マシン - 中期構想 -」、本予稿集 3 B - 5

[3] Ito, N., et al., "The Dataflow-based Parallel Inference Machine To support Two Basic Language in KL1," IFIP-10 Working Conf. on FGCA, Manchester, July, 1985.

[4] 黒川、井田、「リスト処理とアーキテクチャ」、情報処理、23巻、8号、1982