

専攻名		情報理工学	氏名	小川 誠司	指 導	上田 和紀	印
研究指導名		並列知識情報処理	学籍番号	5109B023 – 6 ^{CD}	教 員		
研 究 題 目	階層グラフ書換えモデルを拡張した HyperLMNtal の実現						

1 概要

LMNtal は高い表現力を有する階層グラフ書換えに基づく言語モデルであり, システム検証機能を持つ処理系を備えている. より大規模なシステムの振る舞いを表現するため, 実行性能の向上が進められている. 本研究では LMNtal を階層ハイパーグラフ書換えモデルへ拡張することで, これまで実現困難であった制約記述言語 CHR を始めとしたデータ間に複雑な参照構造を持つ計算モデルを, 従来の LMNtal より自然に記述可能とし, かつ理想的な計算量での実行を実現した. 本発表では HyperLMNtal とそれを用いたモデルの記述例, 動作例などを紹介する.

2 LMNtal とは

LMNtal (Linked Multisets of Nodes transformation language)[2] は基本データ構造のアトム, 一対一の参照構造を表すリンク, 階層を形成する膜からなる階層グラフを, ルールによって書き換えることで計算処理が進む宣言型プログラミング言語である. 現在も表現力の拡張や実行性能の向上に向けた処理系の最適化が進められている. LMNtal ルールの構文は $Head :- Guard \mid Body$ と表記する. $Head$ (ルール左辺) はルールの所属する階層から見た書き換えるべき階層グラフ構造のテンプレートであり, $Body$ (ルール右辺) は書換え後の階層グラフ構造のテンプレートである. また $Guard$ (条件節) を持たせる拡張構文も用意している. ルールが持つヘッド部に合致する部分グラフ構造をグラフ全体から探索する処理をマッチングと呼ぶ.

3 HyperLMNtal への拡張

本研究では新たなデータ構造であるハイパーリンク [3] を LMNtal に導入し, 言語モデルを階層ハイパーグラフ書換えモデルに基づく HyperLMNtal へと拡張した. ハイパーグラフは, 数学におけるグラフを一般化 (拡張) したものであり, エッジ (ハイパーエッジ) が任意個数のノードを連結することができる.

3.1 関連研究: CHR

CHR (Constraint Handling Rules)[1] は制約多重集合の書換えに基づく言語モデルであり, 実アプリケーションへの利用実績もある汎用的な並行制約プログラミング言語である. CHR の構文は膜の無い LMNtal と似通っており, CHR は

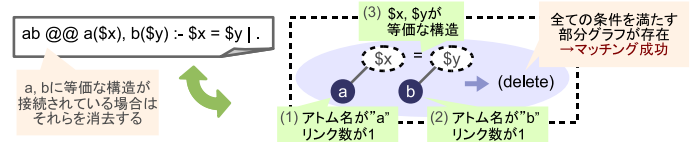


図 1 ルールのマッチング処理

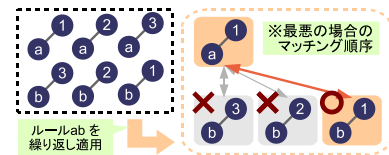


図 2 非効率なマッチングの例

N	10k	20k	40k	80k
best	0.03	0.06	0.10	0.22
worst	1.02	3.98	16.09	66.76

図 3 最良/最悪なマッチングにおける実行時間 [sec]

LMNtal のサブセットであると言えるが, 論理変数を持つために, LMNtal のリンクよりもデータ間の複雑な参照構造を簡潔に表現可能である. また論理変数を利用したマッチング最適化機能も持つため, LMNtal よりも優れた計算量で実行可能な例題もある. CHR を日本語で解説した Web ページ^{*1} が著者によって公開されている.

3.2 背景と目的

LMNtal はルールによって計算処理が進むため, 非効率なマッチングが発生すれば実行時間は著しく低下する. 図 1 は等しい数値を引数に持つアトム a, b の組を見つけるルールのマッチング処理の順序である. このルールを図 2 の左側のグラフに適用させた場合, 図の右側のような冗長なマッチング処理が起こる可能性がある. 初期グラフの数をさらに大きくした場合に起こり得る最良と最悪なマッチング処理では, 図 3 のように実行時間に大きな差がでることから, 非効率なマッチングの防止は LMNtal の実行性能向上に有効であると言える.

LMNtal はリンクによってアトム間の一対一の参照関係を表現する (ハイパーでない) グラフを扱う言語であるが, CHR の論理変数など, データ間に一対一以上の複雑な参照構造を持つ計算モデルを表現するためには, ハイパーグラフに相当する構造を記述する必要がある. 従来の LMNtal では擬似的にハイパーグラフを表現することは可能だが,

^{*1} すっきり CHR, <http://www.ueda.info.waseda.ac.jp/~seiji/trychr/>

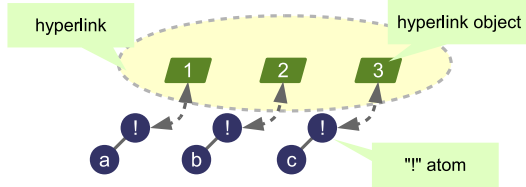


図4 ハイパーリンクの概要図

生成: $Head : -new(X_1), \dots, new(X_i) \mid Atom_1(X_1), \dots, Atom_i(X_i)$
 型制約: $Head : -hlink(X) \mid Body$
 併合: $Atom(H_1), Atom(H_2) :- H_1 > H_2$
 要素数取得: $Atom(H_1) :- H_2 = num(H_1) \mid Body$
 ハイパーリンク主導マッチング: $Atom(\$x), Atom(\$x) :- Body$

図5 ハイパーリンク関連の記法

プログラムの複雑化や非効率的なマッチングの発生を招くために、実用的なハイパーグラフ構造を表現することは不可能であった。

そこで本研究では, LMNtal のリンクを拡張した概念であるハイパーリンクを導入し, CHR の論理変数などの実現困難であった計算モデルを従来よりも自然に記述可能にし, かつ理想的な計算量で実行可能にすることを目指した。

3.3 ハイパーリンクの導入

LMNtal 上でハイパーグラフを表現するため, 新たにハイパーリンクと呼ばれるデータ構造を考案した。図4はハイパーリンクの概要図である。2引数の“!”をファンクタに持つアトム(エクスクラメーションアトム, 以下“!”アトム)に対し, ハイパーリンクを表すオブジェクト(ハイパーリンク構造体)を対応付ける。“!”アトムの第2引数にハイパーリンク構造体へポインタを埋めこんでいる。“!”アトムとハイパーリンク構造体の組をハイパーリンクアトムと呼び, これを処理系内部で Union-Find algorithm に基づく木構造で管理することで, ハイパーリンクを表現する。新たに追加したハイパーリンク関連のガード制約, 演算子などを図5に示す。

4 評価実験

HyperLMNtal の評価実験として, 計7題の CHR プログラムを従来の LMNtal (整数アトム版, 膜版) と HyperLMNtal を用いてエンコードし, それぞれの記述性と実行性能を比較した(図中ではそれぞれ chr, lmn, lmn-mem, lmn-hl)。実行環境のハードウェアは Intel(R) Core(TM)2 Quad 2.60GHz, RAM 4GB, OS は ubuntu 9.10 である。

図6はCHRの例題である不等式制約ソルバであり, これを HyperLMNtal で記述したものが図7である。両者を見比べると HyperLMNtal を用いて CHR の論理変数を自然に表現できていることが分かる。

また図8は Union-Find algorithm を4種類で記法で記述し, その実行時間を比較したものである。lmn は非効率的なマッチングを発生させてしまう記法であるため, 時間計算量が悪化していることが分かる。それ以外の3種類の記法は計算量の面ではほぼ同等の性能であるが, 単純な

$leq(X, X) \Leftarrow true.$
 $leq(X, Y), leq(Y, X) \Leftarrow X=Y.$
 $leq(X, Y), leq(Y, Z) \Rightarrow leq(X, Z).$
 $leq(X, Y) \setminus leq(X, Y) \Leftarrow true.$

図6 不等式制約ソルバの記述例(CHR)

$leq(\$x, \$x) :-.$
 $leq(\$x, \$y), leq(\$y, \$x) :- \$x < \$y.$
 $leq(\$x, \$y), leq(\$y, \$z) \setminus :- uniq(\$x, \$z) \mid leq(\$x, \$z).$
 $leq(\$x, \$y) \setminus leq(\$x, \$y) :-.$

図7 不等式制約ソルバの記述例(HyperLMNtal)

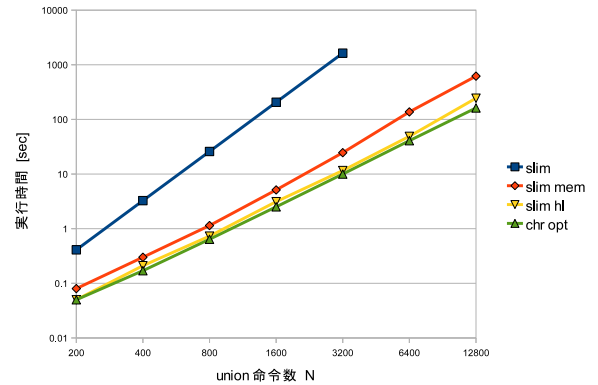


図8 例題(Union-Find algorithm)の実験結果

実行時間では lmn-hl が lmn-mem に優っていることが分かる。またプログラムの文字数とルール数を比較すると, chr, lmn-hl が400字前後であったのに対して lmn-mem は1000字以上であり, ルール数も lmn-mem は他の記法より多い。以上より, HyperLMNtal を用いることでハイパーグラフを従来の LMNtal よりも自然に表現でき, 実行性能の面でもより優れた性能を実現できたと言える。

5 まとめと今後の課題

これまで LMNtal 上では表現困難であったデータ間の多様な参照関係をより自然に記述可能にするために, ハイパーグラフを記述するための新たなデータ構造であるハイパーリンクを処理系に実装した。そして CHR のベンチマーク問題を中心に, 実際にハイパーリンクを利用したプログラムを記述し, 従来よりも理想的な計算量でプログラムの実行が可能になったこと, 例題によっては CHR よりも高速に動作する例題があることなどを確認した。

今後の課題としては, SLIM が持つモデル検査機能への対応, ハイパーリンクへのデータの束縛の検討などが挙げられる。

参考文献

- [1] Frühwirth, T.: Constraint Handling Rules, Cambridge University Press, The Edinburgh Building, Cambridge CB2 8RU, UK, 2009.
- [2] 上田和紀, 加藤紀夫: 言語モデル LMNtal, コンピュータソフトウェア, Vol.21, No.2(2004), pp.126–142.
- [3] 小川誠司, 目黒学, 上田和紀: 階層グラフ書換えモデルを拡張した HyperLMNtal の実現, 第25回人工知能学会全国大会, 2011, 発表予定.