

LMNtalグラフのハイパーグラフへの拡張

小川誠司* 上田和紀**

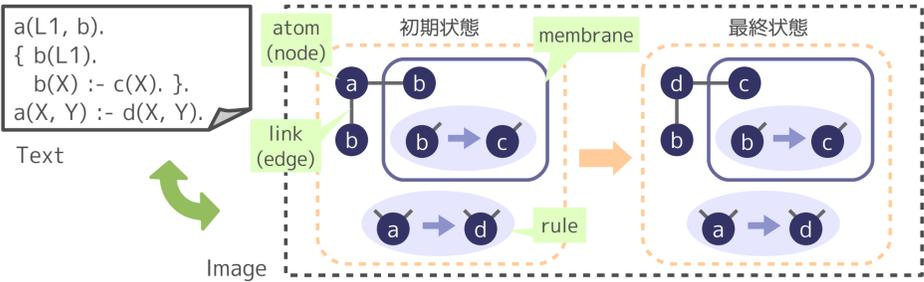
*早稲田大学基幹理工学研究科 **早稲田大学理工学術院

LMNtal (pronounced "elemental") [Ueda 09]

- ◆階層グラフ構造の書換えに基づく並行言語モデル
 - 上田研究室(早稲田大)で開発中の宣言型プログラミング言語
 - 公開URL : <http://www.ueda.info.waseda.ac.jp/lmntal/>

◆LMNtalプログラムの基本構成要素

- アトム(ノード): 基本データ構造
- リンク(エッジ): 2つのアトムを一对一に接続
- 膜 : グラフの階層化、ルール適用範囲の局所化
- ルール : Head(書換え前) :- Guard(条件) | Body(書換え後)

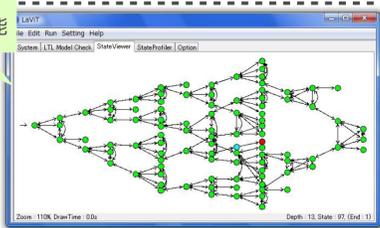


◆複雑なデータ構造の動的変化を簡潔に表現可能

- 様々な計算モデルの統合を実現

シンプルかつ強力な表現力を持つ言語

- ・リアルタイムスケジューリング
- ・分散アルゴリズムの検証
- ・各種探索問題/パズル問題
- ・Coloured Petri Netの変換
- ・Constrain Handling Rulesの変換
- 橋渡し問題
- 水差し問題
- ハノイの塔
- Monkey and Banana
- ナップサック問題



◆LMNtal処理系

- Java版コンパイラ/ランタイム
- SLIM (C言語版ランタイム)
- ・システム検証機能、並列化

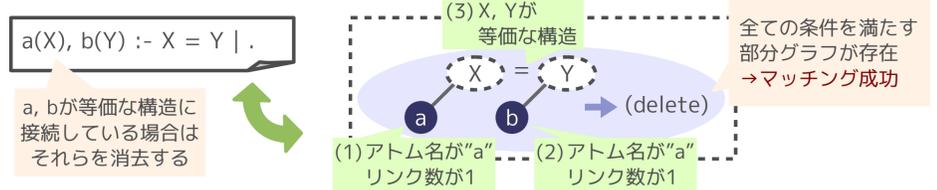
◆処理系の最適化が進行中

- 表現力向上、検証分野への利用拡大

ルールマッチング

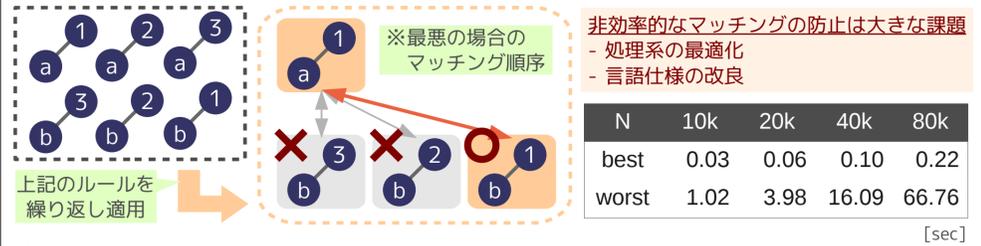
◆ルールマッチング

- 書換え対象となる部分グラフを探索する処理



◆非効率的なマッチングによる実行性能の悪化

- マッチング失敗回数の増加は著しい性能悪化を招く



非効率的なマッチングの防止は大きな課題

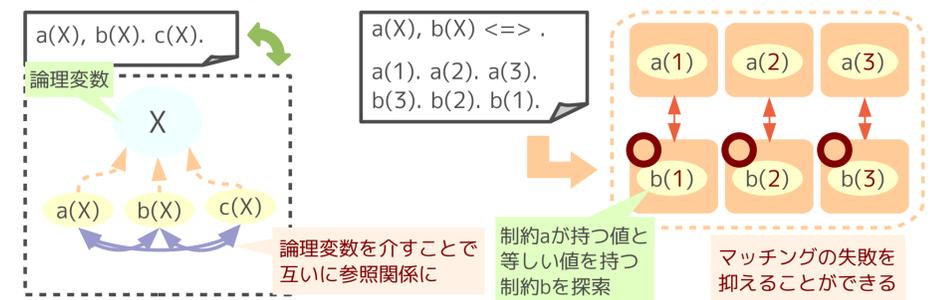
- 処理系の最適化
- 言語仕様の改良

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

[sec]

◆cf. CHR (Constraint Handling Rules) [Chr 09]

- 制約多重集合の書換えに基づく並行言語モデル
- 高い表現力、実アプリケーションへの利用実績
- 膜の無い (フラットな) LMNtalに近い構文
- 論理変数によるデータ間の多様な参照関係を表現可能
- 論理変数を利用したマッチング最適化も存在



LMNtalグラフのハイパーグラフへの拡張

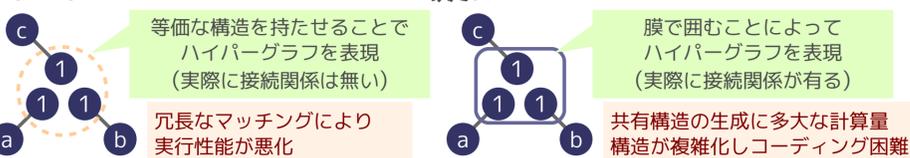
◆LMNtalは(ハイパーでない)グラフを扱う言語

- ハイパーグラフ(エッジ)を表現することは想定外
- リンクは2つのアトムの一对一の参照関係を表す
 - ・ 一对一以上の参照関係を表現困難

cf.ハイパーグラフ

数学におけるグラフを一般化したもの
1本のエッジ(ハイパーエッジ)で任意個数のノードを接続可能

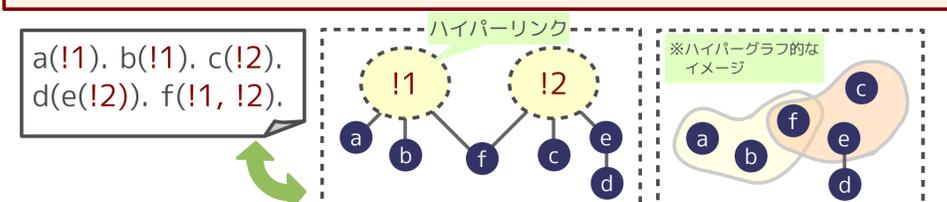
◆従来のLMNtalにおける擬似的なハイパーグラフ



◆LMNtal処理系へのハイパーリンク機能の導入

- 1本のリンクで任意個数のアトムを接続可能
- CHRを自然な形でエンコード可能

LMNtalグラフのハイパーグラフへの拡張を実現 論理変数などの多様な参照関係を従来より自然に表現可能に

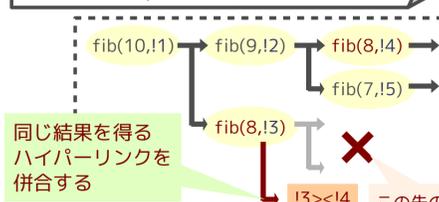


◆ハイパーリンクを用いたマッチング最適化

- ハイパーリンクの接続関係を活用
 - ・ 非効率的なマッチングを防ぐプログラミングが可能
 - 異なるハイパーリンク同士の接続 (併合)
 - ・ 再帰的な計算処理の簡略化が可能

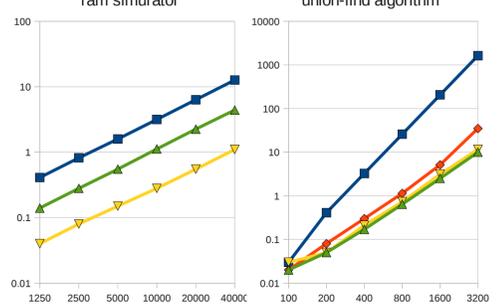
fibonacci number

```
unify @@ fib($n, N1, M1) \ fib2($n, N2, M2) :-
    hlink(M1), int(N2) | M1 >> M2.
fib1 @@ fib2(N,0,M) :- hlink(N) | v(M, 1).
fib2 @@ fib(N,1,M) :- hlink(N) | v(M, 1).
fib3 @@ fib($n, N, M) :-
    N > 1, N1 = N-1, N2 = N-2,
    make(N, $a), make(N1, $b),
    new($x), new($y), hlink($n) |
    fib($a, N1, $x), fib2($b, N2, $y), M = $x + $y.
add @@ v($x, X), v($y, Y), H = $x + $y :-
    Z = X+Y | v(H, Z), v($x, X).
```



CHRよりも高速に実行可能

従来のLMNtalよりも理想的な計算量で実行可能



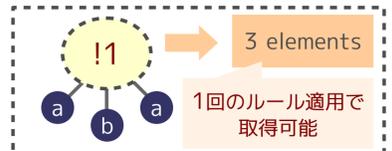
実験環境
CPU : 2.6GHz, Intel Core2 Quad
RAM : 4.0GB
OS : ubuntu 9.10

◆要素数の取得

- LMNtalでは特定の要素数を取得する方法が無かった

◆今後の課題

- より大規模な例題のエンコード
- システム検証機能への対応



[Ueda 09] Kazunori Ueda: LMNtal as a Hierarchical Logic Programming Language, Theor. Comput. Sci. vol.410, pp.4784-4800, 2009.

[Chr 09] Thom Frühwirth, "Constraint Handling Rules", 2009, Cambridge University Press, The Edinburgh Building, Cambridge CB2 8RU, UK.