人　工　知　能 91－1
（1993．11．24）

結合グラフ証明手続きにおける効率的な探索戦略

中嶋 卓雄，國安 治，中村 良三
熊本大学 工学部

本論文では，結合グラフを構成するリンク数およびリテラル数に重みを付けた評価関数による効率的な探索戦略を提案する．

提案する探索戦略では，探索空間の広さを表す結合グラフのリンク数と探索空間の大きさを表すリテラル数に基づいた評価関数を導入している．さらに，結合グラフの2連結成分を考慮しグラフの構造的な変化に注目した探索戦略について考察する．提案する評価関数に基づく探索戦略およびトップダウン・ボトムアップなどの探索戦略を推論ステップ毎に切り替えることができる定理証明システムをXウィンドウ上にModula-2で具現化した．このシステムにより提案する探索戦略を評価する．

An Efficient Search Strategy for the Connection Graph Proof Procedure

Takuo Nakashima, Osamu Kuniyasu, Ryozo Nakamura

Faculty of Engineering, Kumamoto University
2–39–1, Kurokami, Kumamoto-shi ,860 ,Japan

This paper presents an efficient search strategy baced on an estimation function obtained from the structure of a connection graph. The estimation function grows in weight of the number of links and literals. The proof system in which this new strategy is included has been implemented in Modula-2 on X window system. The performance of this strategy is also compared by this proof system.

1 はじめに

Kowalski[1] によって提案されたレゾリューションに基づく結合グラフ証明手続きは他の証明手続きと比較しても，効率的に探索空間を抑えることができる．

結合グラフ証明手続きにおいて，効率に影響する要素は大きく 2 つ存在する．一方は，結合グラフから任意に 1 つのリンクを選択する戦略であり，他方は，冗長な節およびリテラルを削除する規則である．

これまで，結合グラフ証明手続きを効率化する試みとして，変数が単一化可能な領域 (domain) を計算し，その情報を基に新たな節の削除規則 (υ−*rule*) が導入されたり [3]，論理的な無矛盾性を保持する新らしい包含規則が提案されてきた [4]．これらの提案はおもに節を削除する規則に注目したものである．また，結合グラフの節形式を限定し，簡単にテストできるプリミティブなリテラルの存在を前提として，そのプリミティブなリテラルを条件としてリンクに付加した新たな結合グラフ証明手続きが提案されている [6]．しかし，証明を方向付けるのは，リンクを選択する戦略であり，節形式を限定しないで，一般的な非ホーン節による探索戦略を考察するのは重要である．

従来，結合グラフ証明手続きにおいて，非決定的に 1 つのリンクを選択する戦略には，結合グラフ全体のリンク数が最も減少するリンクを選択する方法 [3] が用いられきた．また，レゾリューションに基づく一般的な証明手続きにおいては，節を構成するリテラルやリテラル中の項に重みを付けた評価関数による発見的な探索戦略が提案されている [5]．

本稿では，探索空間の広さを表す結合グラフのリンク数と探索空間の大きさを表すリテラル数に基づいた評価関数を導入した発見的な探索戦略を提案する．さらに，結合グラフの 2 連結成分を考慮して，グラフの構造的な変化に注目した探索戦略について考察する．

また，提案する評価関数に基づく探索戦略およびトップダウン・ボトムアップなどの探索戦略を推論ステップ毎に切り替えることができる定理証明システムを Modula-2 により具現化した．このシステムは GUI として X ウィンドウシステムを採用しており，探索戦略のシミュレーションも容易である．

まず，2 章では，結合グラフ証明手続きを概観する．3 章では，リンク数とリテラル数から構成される評価関数による探索戦略と，2 連結成分を考慮してグラフの構造的な変化に注目した探索戦略を提案し，簡単な削除規則も示す．4 章では，具現化した定理証明システムの機能を紹介し，5 章では，そのシステムにより提案した探索戦略を評価する．

2 結合グラフ証明手続き

本稿で用いる用語は Kowalski の文献 [2] の定義に従うものとする．また，Munch の文献 [3] に準じて次のように還元規則を表す．

π−rule ピュアリテラル (pure literal) を含む節を削除する．

ϱ−rule リンクを対応するレゾルベントによって置き換える．

τ−rule トートロジー (tautologie) な節を削除する．

σ−rule 包含される (subsumed) 節を削除する．

ε−rule ファクタリング (factoring) を適用する．

結合グラフ証明手続き

結合グラフ証明手続きでは，まず，与えられた節集合から初期結合グラフを生成する．

たとえば，図 1(a) に表した節集合から (b) に示す初期結合グラフが得られる．

次に，その結合グラフにレゾリューションおよび削除規則を適用することによりグラフを解き，グラフから節がなくなれば無矛盾が証明され，空節が導かれれば矛盾が証明されて手続きが終了する．

前述した還元規則を使うと，結合グラフ証明手続きは次のように定義される．

Step.1 与えられた節集合から初期結合グラフを作成する．

Step.2 π−rule，τ−rule，σ−rule およびε−rule を適用し節を削除する．さらに，リンクでつながれていないリテラルを含む節とその節に連結されているリンクをすべて削除する．

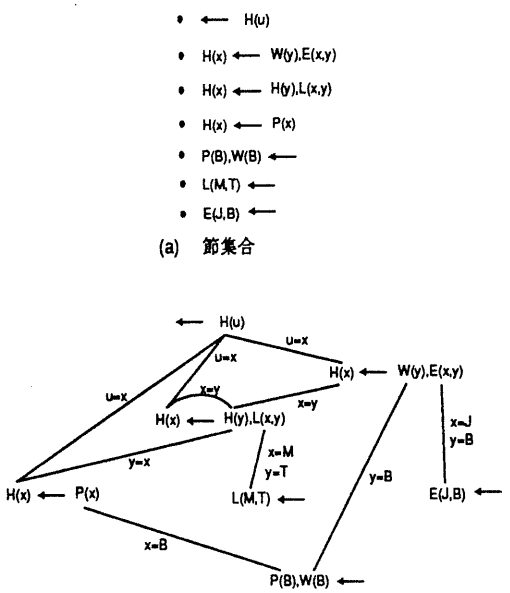


図 1 節集合とその初期結合グラフ

Step.3 グラフが空なら証明は失敗し，グラフが空節を含むならば証明が成功し，手続きが終了する．

Step.4 ϱ–rule を適用する．すなわち疑似 (psudo) リンクではないリンクを選択し，リンクの両端となる親の節からレゾルベントを生成し，グラフに追加し，選択したリンクを削除する．

Step.5 レゾルベント中の各リテラルについて，対応する親の節から連結している相補なリテラルとの単一化が可能ならばリンクを張る．

Sterp.6 Step.2 へ戻る．

3 提案する探索戦略

従来から，結合グラフ証明手続きにおけるリンクの選択には，発見的な戦略が用いられており，選択による 1 回のレゾリューションのあと，結合グラフ中のリンク数が最も減少するリンクが選択されてきた [3].

しかし，リンクの数が同じであっても，必ずしも同数の推論ステップで証明が終了とは限らない．

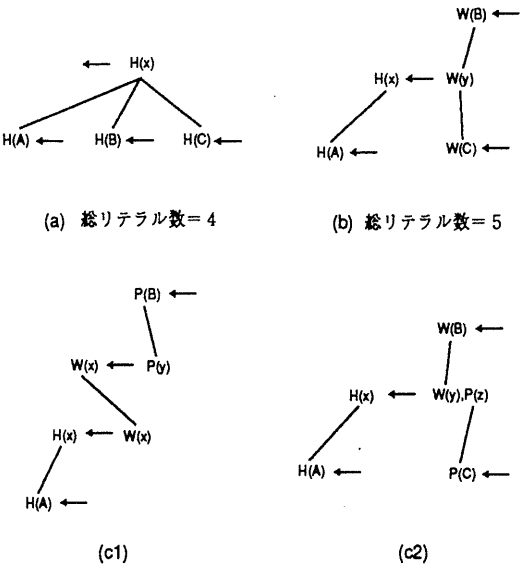


図 2 リテラル数が異なる結合グラフの例

たとえば，図 2に示すようにリンク数が同数であっても，(a) では 1 回，(b) では 2 回，(c) では 3 回の推論ステップが必要となる．

本章では，リンク数だけではなく，リテラル数も考慮した発見的な探索戦略を提案する．

結合グラフのモデル化

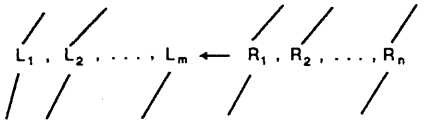
結合グラフをリンク数とリテラル数から抽象化した無向グラフにモデル化する．

1. 結合グラフのリンクを無向グラフの辺とする．
2. 結合グラフの節を無向グラフの頂点とする．なお，頂点に結合グラフの節に含まれるリテラル数を記入する．

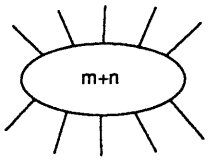
たとえば，次の図 3(a) に示す結合グラフの 1 つの節は (b) のようにモデル化される．

評価関数

提案する評価関数は，結合グラフの探索空間を数量化し，あるリンクを選択し，1 回推論することによって減少する探索空間が最大となるような関数を構成する．



(a) モデル化する節



(b) モデル化された節点

図 3 モデル化した結合グラフ

まず，結合グラフの探索空間は，広さを表すリンク数（G_{link}）と大きさを表すリテラル数（$G_{literal}$）によって構成されると考える．1 回の推論によって，減少するリンク数をΔG_{link}，リテラル数を$\Delta G_{literal}$と表すと，あるリンク（$aLink$）の評価関数（$F(aLink)$）を次のように表し，結合グラフ証明手続きでは評価関数の値が最大となるリンクを選択する．

$$F(aLink) = W_{link} \times \Delta G_{link} + W_{literal} \times \Delta G_{literal} \quad (1)$$

ここで，W_{link}はリンク数に対する重みであり，$W_{literal}$はリテラル数に対する重みである．

たとえば，図 1に示した初期結合グラフをモデル化すると次の図4のようになる．ここで，リンクの評価をするため，各リンクに識別番号を付加する．

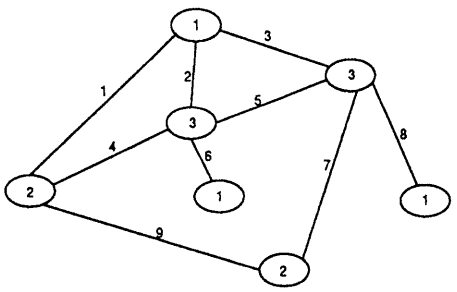


図 4 結合グラフのモデル化の例

このとき評価関数は次のように算出できる．

$$\begin{aligned}
F(1) &= W_{link} \times 0 + W_{literal} \times (-1) \\
F(2) &= W_{link} \times (-3) + W_{literal} \times (-2) \\
F(3) &= W_{link} \times (-1) + W_{literal} \times (-2) \\
F(4) &= W_{link} \times (-3) + W_{literal} \times (-3) \\
F(5) &= W_{link} \times (-4) + W_{literal} \times (-4) \\
F(6) &= W_{link} \times 1 + W_{literal} \times 2 \\
F(7) &= W_{link} \times 1 + W_{literal} \times 2 \\
F(8) &= W_{link} \times 1 + W_{literal} \times 2 \\
F(9) &= W_{link} \times 1 + W_{literal} \times 2
\end{aligned}$$

この評価関数を得るには選択したリンクの親の節の単一化状態を把握すれば十分である．したがって，手続きを実行する上であまり負荷とはならない．

関節点に注目した評価関数

次に，結合グラフの 2 連結成分を考慮してグラフの構造的な変化に注目した探索戦略について考察する．

関節点に注目した結合グラフは次のような意味的な構造を持つとする．

- 証明手続きは結合グラフのすべての節を利用する．
- 結合グラフの意味的なつながりは結合グラフの 2 連結成分に対応している．
- 関節点は意味的なつながりを統合する役割を持つ．

証明手続きは意味を統合しながら，意味的なつながりを単純化していく操作であると考えて，次のような順に高い優先度を付けてリンクを選択する．なお，各優先度の操作において複数の選択枝がある場合，前述した評価関数によりリンクを選択する．

1. 関節点が通常の節点になり 2 つの 2 連結成分が 1 つになるリンクを選択．
2. 2 連結成分が 1 つになるリンクがない場合は，最も節点数が少ない連結成分に対して以下の順に操作する．
   (a) 関節点を含む閉路が存在するときには，その閉路がなくなるリンクを選択．

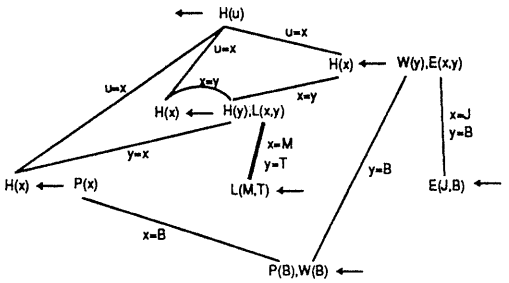
(b) もし，閉路がなくならない場合，閉路中のリテラル数が最も減少する閉路のリンクを選択する．

削除規則

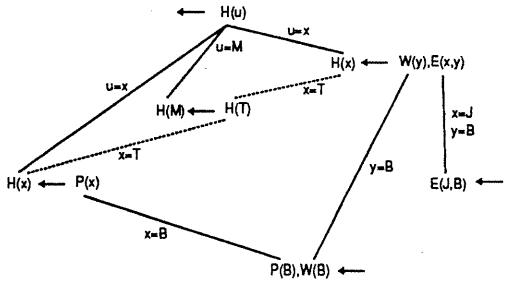
リンクに保持している変数に対する代入を利用した新たな削除規則を提案する．

この規則は前述した結合グラフ証明手続きの **Step.2** において実行される．

削除規則 生成されたレゾルベントに親節から引き継いだリンクを接続するとき，変数に定数または関数の代入が生じるならば，その代入が接続先の節の中で変数を共有するリテラルのリンクにおいて代入が競合しないか調べる．もし競合するならばそのリンクは生成しない．



(a) リンク選択前の結合グラフ



(b) 削除規則が適用されている結合グラフ

図 5 結合グラフに適用された削除規則の例

たとえば，図 1の初期結合グラフにおいて，図 5(a) の太線で示すリンク

$$L(M,T) \leftrightarrow L(x,y)$$

を選択すると，レゾルベント $H(M) \leftarrow H(T)$ が生成される．$H(T)$ とリンクする可能性がある 2 つの節のうち，$H(x) \leftarrow W(y), E(x,y)$ に対して，すでに変数 x が $x = J$ としか単一化する可能性が残されていないので，リンクは生成されず，もう一方の節 $H(x) \leftarrow P(x)$ に対しても，すでに変数 x が $x = B$ としか単一化する可能性が残されていない．したがって，図 5(b) に破線で示すリンクは生成されず，レゾルベントも生成されない．

4 定理証明システム

結合グラフの具現化については，並列結合グラフ証明手続きの効率化として項のデータ構造を工夫したり [7]，異なるタイプの並列化を効率よく具現化する [8] などが提案されている．しかし，リンク選択に関する探索戦略については詳細化されていない．

前述した評価関数に基づく探索戦略およびトップダウン・ボトムアップなどの探索戦略を推論ステップ毎に切り替えることができる定理証明システムを具現化した．本章では，その構造および機能について説明する．

具現化したシステムは Modula-2 で記述し，GUI として X ウィンドウシステムを採用している．図 6に定理証明システムのウィンドウを示す．

図 6に示すように，このシステムは 3 つのパネルから構成される．

操作パネル 図 6の上部に配置しているパネルで，ファイルの入力 (Load ボタン)，ファイルの編集，出力 (Edit ボタン)，ゴールの入力 (Goal ボタン) などの基本的な編集ボタンを持ち，さらに評価関数の重み (W_{link}, $W_{literal}$) の設定およびクリア (Reset ボタン)，推論戦略の切り替えなどが可能である．また，マニュアルで任意にリンクを選択できるように，リンク選択番号を入力するフィールドを設けている．

コンソールパネル 図 6の中央部分に配置している

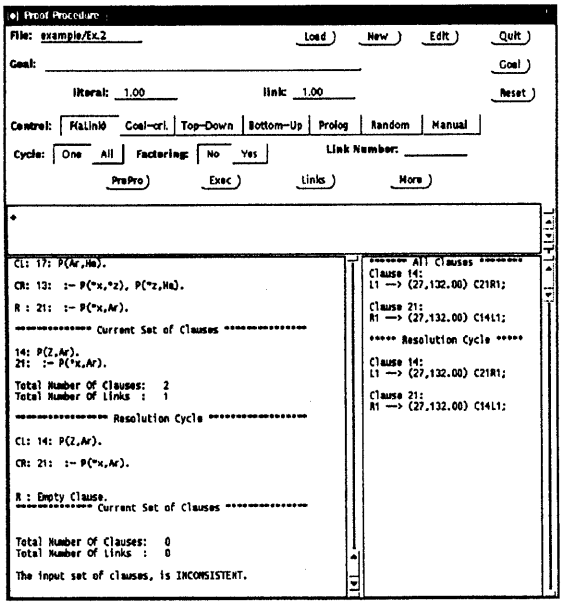


図 6 定理証明システム

パネルで，構文エラーおよびシステムメッセージを表示している．

結合グラフ表示パネル 図 6の下部左側に配置しているパネルで，節に対して入力された順および新しく生成された順に識別番号を付け，現時点での結合グラフの節集合を表示している．

リンク表示パネル 図 6の下部右側に配置しているパネルで，リテラルが持つリンクの識別番号，評価関数の値およびリンクがつながっている相補なリテラルの節識別番号とリテラル識別番号を示している．

実行制御

評価関数 この定理証明システムは基本的にすべて評価関数を計算することにより実行を制御している．

たとえば，Prolog 処理系のようなトップダウン実行を実現するために，次のような規則に基づく評価関数を構成している．

1. ゴール節に結ばれているリンクを選択する．
2. ゴール節の左のリテラルから順にリンクの優先度を下げる．
3. 入力された節の正のリテラルについて，入力順にリテラルのリンクの優先度を下げる．

探索戦略の切り替え この定理証明システムはインタープリタとして推論ステップ毎に動作する．次のような探索戦略が可能である．

F(L,R) 提案した評価関数に基づくリンクの選択．

Top-Down 負のリテラルからのみ構成される節のリンクを選択．非決定的な選択に関しては提案した評価関数を用いる．

Bottom-Up 正のリテラルのみから構成される節のリンクを選択．非決定的な選択に関しては提案した評価関数を用いる．

Prolog Prolog と同じ計算規則，選択規則に基づきリンクを選択．

Random ランダムにリンクを選択．

Manual 手動でリンクを選択．

5 評価

図 1に示す例と付録に示す例 1 から例 5 までの節集合に対して結合グラフを構成し，提案する評価関数に基づいて推論を実行した．表 1 には，Prolog の探索戦略と提案する探索戦略における推論ステップを示す．特に提案する探索戦略については W_{link} と $W_{literal}$ が異なる場合の推論ステップ数を示す．

表 1から，Prolog と比較して結合グラフ証明手続きが非常に効率的であるのは明白である．また，リテラル数を考慮しない場合（$W_{literal} = 0$）と比べ，考慮した場合の方が合計の推論ステップが減少しているのが分かる．

6 おわりに

本論文では，結合グラフを構成するリンク数およびリテラル数に重みを付けた評価関数による発見的な探索戦略を提案し，複数の戦略を実行できる定理証明システムを具現化した．

表 1 推論ステップ数の比較

	$W_{link}=1$ $W_{literal}=0$	$W_{link}=1$ $W_{literal}=1$	Prolog
例	6	5	–
例 1	3	3	4
例 2	10	9	6
例 3	8	8	8
例 4	10	9	13
例 5	12	12	17
例 6	19	20	21
合計	68	66	68
平均	9.7	9.4	11.3

提案した探索戦略は探索空間を適切に表す尺度によって構成され，さらにその計算も容易である．また，推論過程における結合グラフの構造的な変化に注目した探索戦略についても考察した．

今後は，Munch[3] などが提案しているように，ドメインを変数に保持させることによって，削除規則を広範囲に適用するなどの効率化が必要である．

参考文献

[1] R.A. Kowalshi, "A Proof Procedure Using Connection Graphs" *J ACM*, 22, pp. 572–595, 1975.

[2] R.A. Kowalski, *Logic for Problem Solving*, Artificial Intelligence Series, Vol.7, North-Holland, New York, 1979. 浦昭二監修，山田眞市，菊池光昭，桑野龍夫訳，論理による問題の解法，培風館，1987

[3] K.H. Munch, "A new reduction rule for the connection graph proof procedure" *J. Autom. Reasoning(Netherlans)*, Vol. 4, No. 4, pp. 425–444, Dec., 1988.

[4] B.M. Kim and J.W. Cho, "A new subsumption method in the connection graph proof procedure" *Theoretical Computer Science*, Vol.103, No.2,pp.283–309,1992.

[5] C.L. Chang and R.C.T. Lee *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, New York, 1973.

[6] J.Barklund,N.HagnerandM.Wafin, "Condition graphs" *Logic Programming: Proceeding of the fifth International Conference and Symposium*, Vo.1–2, pp.435–446, 1988.

[7] D.M.W. Powers, L. Davila and G. Wrightson, "Implementing Connection Graphs for Logic Programming", *Proceedings of the ninth Europeaon Cybernetics and Systems Research*, Vol.2, pp.957–964, 1988.

[8] D.M.W. Powers, "Parallel and Efficient Implementation of the Compartmentalized Connection Graph Proof Procedure", *Parallelization in Inference Systems: International Workshop Proceedings*, pp.210–233, 1992.

付録

[例 1]

$$
\begin{array}{lcl}
F(x) & \leftarrow & H(x) \\
G(S) & \leftarrow & \\
H(T) & \leftarrow & \\
H(S) & \leftarrow & \\
 & \leftarrow & F(x), G(x)
\end{array}
$$

[例 2]

$$
\begin{array}{rcl}
F(Z, Ar) & \leftarrow & \\
M(He, Ar) & \leftarrow & \\
F(Ar, Ha) & \leftarrow & \\
M(Ap, Ha) & \leftarrow & \\
F(C, S) & \leftarrow & \\
M(Ha, S) & \leftarrow & \\
F(Z, D) & \leftarrow & \\
M(S, D) & \leftarrow & \\
P(x, y) & \leftarrow & M(x, y) \\
P(x, y) & \leftarrow & F(x, y) \\
G(x, y) & \leftarrow & P(x, z), P(z, y) \\
 & \leftarrow & G(u, Ha)
\end{array}
$$

[例 3]

$$
\begin{aligned}
T(E,A) &\leftarrow \\
A(T,Y) &\leftarrow \\
J(Y,K) &\leftarrow \\
K(E,Y) &\leftarrow \\
K(A,Y) &\leftarrow \\
K(K,T) &\leftarrow \\
J(x,y) &\leftarrow A(x,z), K(y,z) \\
J(x,y) &\leftarrow T(x,z), K(y,z) \\
A(x,y) &\leftarrow T(y,x) \\
Y(x,y) &\leftarrow K(y,x) \\
I(z,y) &\leftarrow J(x,y), Y(x,z) \\
&\leftarrow I(A,K), I(E,K)
\end{aligned}
$$

[例 4]

$$
\begin{aligned}
U(A,B) &\leftarrow \\
W(A,C) &\leftarrow \\
X(A,x) &\leftarrow \\
S(C) &\leftarrow \\
P(x) &\leftarrow R(x,y), S(y) \\
Q(x) &\leftarrow T(y,x) \\
T(x,y) &\leftarrow P(x), U(x,z) \\
T(x,y) &\leftarrow V(x,y), X(x,y) \\
V(x,y) &\leftarrow W(x,y) \\
R(x,y) &\leftarrow X(x,y), W(x,y) \\
&\leftarrow P(A), Q(B)
\end{aligned}
$$

[例 5]

$$
\begin{aligned}
The(A,B) &\leftarrow \\
girl(B,C) &\leftarrow \\
guides(C,D) &\leftarrow \\
fish(D,E) &\leftarrow \\
det(x,y) &\leftarrow The(x,y) \\
noun(x,y) &\leftarrow girl(x,y) \\
vb(x,y) &\leftarrow guides(x,y) \\
noun(x,y) &\leftarrow fish(x,y) \\
np(x,z) &\leftarrow det(x,y), noun(y,z) \\
np(x,y) &\leftarrow noun(x,y) \\
vp(x,y) &\leftarrow vb(x,y) \\
vp(x,z) &\leftarrow vb(x,y), np(y,z) \\
S(x,z) &\leftarrow np(x,y), vp(y,z) \\
&\leftarrow S(A,E)
\end{aligned}
$$

[例 6]

$$
\begin{aligned}
The(A,B) &\leftarrow \\
girl(B,C) &\leftarrow \\
guides(C,D) &\leftarrow \\
fish(D,E) &\leftarrow \\
det(x,y) &\leftarrow The(x,y) \\
adj(x,y) &\leftarrow girl(x,y) \\
noun(x,y) &\leftarrow girl(x,y) \\
vb(x,y) &\leftarrow guides(x,y) \\
noun(x,y) &\leftarrow guides(x,y) \\
vb(x,y) &\leftarrow fish(x,y) \\
noun(x,y) &\leftarrow fish(x,y) \\
np(x,z) &\leftarrow det(x,y), noun(y,z) \\
np(x,v) &\leftarrow det(x,y), adj(y,z), noun(z,v) \\
np(x,y) &\leftarrow noun(x,y) \\
vp(x,y) &\leftarrow vb(x,y) \\
vp(x,z) &\leftarrow vb(x,y), np(y,z) \\
S(x,z) &\leftarrow np(x,y), vp(y,z) \\
&\leftarrow S(A,E)
\end{aligned}
$$