

イネの穂の分枝構造を図示する Prolog プログラム

石川哲也*・鈴木保宏

(農業技術研究機構)

要旨：米粒の成分含量や物性を1粒単位で測定することが最近可能となり、これらの測定値と穂上の着生位置との関係を明らかにするための研究が進められている。そこで、イネの穂の分枝構造を模式的に図示し、個々の穎花の持つ情報を穂上の着生位置に対応させて表示するためのプログラムを作成し、その有用性について検討を行った。穂の分枝構造をリスト形式で記述することにより、その多様性に対応することが可能となった。描画プログラムの作成には、「パターン照合」や「単一化」などの特色を有するプログラム言語 Prolog を採用し、再帰的呼び出し機能を活用した。表形式で記録された既存の分析結果を有効に利用するため、それらをリスト形式のデータに変換するプログラムを作成した。さらに、穎花の現存・退化といった定性的情報の表示に加えて、新鮮重など定量的情報に対応した塗り分け機能をプログラムに追加した。描画された模式図は、枝梗の長さや穎花の形状という観点からは正確ではないが、枝梗や穎花の相対的な位置は正しく表現されており、穎花の定性的・定量的特性と着生位置との関係を直観的に把握する一助となる。さらに、この描画プログラムは、同様の分枝構造を持つイネ属の穂や、イネの分けつ体系等の描画への応用も可能である。

キーワード：イネ、再帰的呼び出し、Prolog、分枝構造、穂、リスト。

イネの穂では一般的には穂軸から1次枝梗が分枝し、1次枝梗の先端側には穎花が直接着生するが、穂軸に近い基部側には2次枝梗が分枝する場合がある。2次枝梗には通常は穎花のみが直接着生するが、品種・栽培条件によっては3次枝梗の分枝が認められる場合がある(福嶋 1999 a, b, 石川ら 1999)。穂の分枝構造が穎果の発育に及ぼす影響については、受精後の穎果の発育が穂上の位置により異なること、穂上には強勢穎果、弱勢穎果が分布し、穂への転流物質が不足するとき穎果間で競合が生じることが示されている(荒井・河野 1978, 1979)。玄米の外観品質では、穂上での着生位置と心白、腹白や乳白等との関係(木戸・梁取 1968, 田代・江幡 1974)が検討されているが、各種成分含量や物性の分析には、穂を上部・中央部・下部に区分したり、1次枝梗と2次枝梗に着生する穎果を区別する(Matsueら 1994)など、穎果をグループ化して一定のサンプル量を確保し、その部分の平均的な特性として検討されている。近年、米粒の成分を1粒単位で測定することが可能となり、アミロース、窒素および無機成分の含量も着生位置により変動することが明らかにされている(堀内 1991, 鈴木ら 1994, 進藤ら 1998)。このようにして得られた個々の穎花・米粒の持つ定量的・定性的な情報を、穂上の着生位置と関連づけるためには、穂の分枝構造を図示し、個々の穎花の着生位置に対応させてその情報を表示する必要がある。このような表示は、実際の穂を模写する方法(進藤ら 1998)や模式図で示す方法(堀内 1991)があるが、ともに時間と労力を要する。

筆者らは、栽培条件が穂の着粒構造に及ぼす影響とその品種間差に関する研究を進めている(石川ら 1998)が、個体変動を含んだ着粒構造の調査結果は、品種固有の特性や栽培条件の影響などが直観的に認められるにもかかわらず、

統計的に処理することが困難な場合がある。したがって、穂の着粒構造を統一的な書式で管理するとともに、個々の穂の分枝構造を模式図として示し、同時に着生する穎花の定性的・定量的情報を表示して、品種・処理間差を比較することが重要と考える。そこで、穂の分枝構造の記述形式を示すとともにその結果を模式的にコンピュータ画面に描画し、さらに個々の穎花の持つ情報を穂上の着生位置に対応して表示するためのプログラムを作成し、その有用性について検討を行ったので報告する。

プログラムの作成と実行結果

1. 分枝構造を記述するためのデータ形式

穂の分枝構造を記述するためのデータ形式として、「リスト」を採用した。リストは中カッコ“[]”の中に、穎花の現存・退化などを表す定性的情報や、新鮮重・窒素含有率などを表す定量的情報である「要素」をカンマ“,”で区切って並べるデータ形式である(黒川 1985)。BASIC や FORTRAN などの一般的なプログラム言語には「配列」というデータ形式があり、2次枝梗まで分化している穂の分枝構造を記述する場合には、3次元の配列を用意すればデータを格納することは可能である。しかし、配列の領域は、

1) 予想される最大の要素数(1次枝梗の本数など)を想定して確保するため、1次枝梗ごとに着生穎花数や2次枝梗数が異なると、使用されない部分が生じる、

2) 予想される最大の次元(3次枝梗があるかどうかなど)を想定して確保するため、2次枝梗や3次枝梗の本数があまり多くない場合、配列のかなりの部分が使用されない、

という欠点を有する。

これに対して、リスト形式が分枝構造の表記に適している理由として、

1) 要素数は可変であるため、枝梗ごとに要素数が異なってもかまわない、

2) 要素が一つもない「空」リストの存在も可能であり、これは退化して痕跡のみ残っている枝梗と対応づけることができる、

3) 要素としてリスト自身を含むことが可能であるため、2次枝梗や、一部の品種で認められる3次枝梗にも対応可能である、

の3点を挙げることができる。このようなリスト形式の特徴により、遺伝的・環境的要因が作用して多様な分枝構造を示すイネの穂を容易に記述することができる。以下、本報告では、データ形式としてのリストを「リスト」、分枝構造をリスト形式で記述したデータを「リスト・データ」と区別して表記する。

リストでは不定個の要素を含むことができるが、含まれる要素の数はリストの終端まで順に数えて求める必要がある。分枝の基部側から先端に向かって要素を記録する方法でリスト・データを作成する場合には、分枝位置から始めて一定の長さだけ分枝を延ばしてから要素を描画することにより、任意の長さに対応できる。すなわち、リスト・データに含まれる要素の数をあらかじめ求めておく必要はなく、プログラムが簡潔となる。これに対して、分枝の先端から基部側に向かってリスト形式で記述する場合には、

1) 分枝位置から描画するには、リスト・データを逆順にする、

2) 先端から描画するには、あらかじめリスト・データの要素の数を数えて、模式図における分枝の先端の座標を算出する、

のいずれかの処理を要する。そこで、本プログラムでは、穂軸および分枝の基部側から分枝構造を記述した。

リスト形式で分枝構造を記述するとき、穎花を表す要素は空リスト以外であればよい。たとえば、穎花の現存・退化という定性的な情報を記録する場合は、それぞれ“o”、“x”という文字に対応させることができる。この方法を用いて、リスト形式で仮想的な1次枝梗を記述した結果を第1表に示した。

2. プログラムの基本構成

このようにして定義された分枝構造のリスト・データを処理するため、プログラム言語 Prolog を採用した。Prolog はリスト処理に優れる他に、データベースからの検索を行う「パターン照合」や、変数への代入と比較を同時に行う「単一化」など他の言語にない考え方が含まれている（黒川 1985, フィリピッチ 1990）ため、作成されたプログラムは簡潔で可読性が高い特徴を有する。Prolog はさまざまなオペレーティング・システムで動作するものが開発されているが、本報告では、ダブリン大学トリニティ

・カレッジの Michael Brady により開発され、Mac OS 7.5.5 以降および Mac OS X のクラシック環境で動作し、グラフィック処理系を持つ Macintosh 用の Open Prolog を使用した。Open Prolog は、開発者の開設したインターネット上のホームページ (<http://www.cs.tcd.ie/open-prolog/> 最終更新日 2003/3/24) で入手可能であり、約 1.2 MB のディスク容量と 4 MB 以上のメモリ空き領域を必要とする。

本プログラムにおける処理の流れを第1図に示した。穂の分枝構造を記述したリスト・データをプログラムに入力すると、以下のように処理が進行する。

1) リスト・データから先頭の要素を取り出す。穂全体を記述したリスト・データからは、もっとも穂首側から分枝している1次枝梗を表すリスト・データが取り出される。1次枝梗を表すリスト・データからは、もっとも基部側から分枝している2次枝梗を表すリスト・データ、または直接着生している穎花のデータが取り出される。

2) 取り出された要素が分枝を表すリスト・データである場合は、本プログラムを再帰的に呼び出し、入力データとして与える。

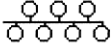
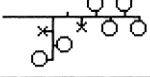
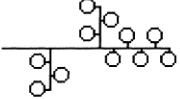
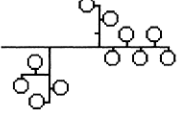
3) 取り出された要素が現存・退化穎花の場合は、対応する図形を描画する。

4) 1) で先頭の要素を取り除いたリスト・データに別の要素が残っている場合には、本プログラムを再帰的に呼び出して、先頭の要素を取り除いたリスト・データを与える。これにより、元のリスト・データの2番目の要素が処理される。この再帰的呼び出しを、リスト・データが空になるまで繰り返して実行することにより、すべての要素を処理することができる。

このように、分枝を表すリスト・データの階層構造によって穂全体の分枝構造を記述したことにより、同一のプログラムを再帰的に呼び出して、穂全体、1次枝梗、2次枝梗の順に、理論的にはどの次数までも処理することが可能である。

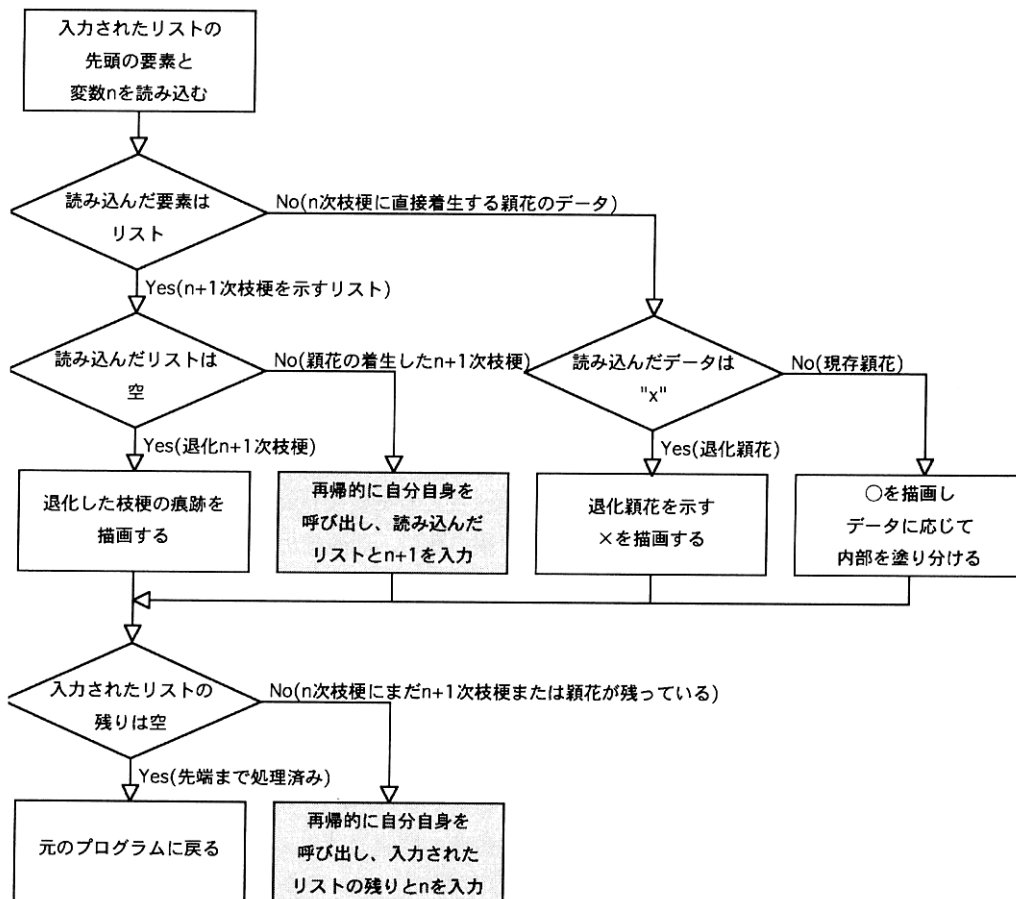
本プログラムは3次枝梗まで描画することを目的としているが、グラフィック画面の広さは限られているので、模式図の一部が重なったり、分枝の間隔が間延びすることを避ける必要がある。そのため、現在処理しているリスト・データが穂軸か、 n 次枝梗かを示す変数 n （穂全体の場合は $n=0$ ）を導入して、 n の値に応じて分枝や穎花の着生位置の間隔のパラメータを調整した。上述したプログラムの流れ1)~4)のうち、2)における再帰的呼び出しでは変数 n の現在の値に1を加えて引き渡し、4)における再帰的呼び出しでは変数 n の値をそのまま引き渡す。これにより、1次枝梗を表すリスト・データを処理しているときは $n=1$ 、2次枝梗を表すリスト・データを処理している時は $n=2$ となり、それぞれ該当する分枝間隔のパラメータが利用される。このように、再帰的呼び出しを活用することにより、簡潔なプログラムの作成が可能となった。

第1表 7つの要素が着生する仮想的な1次枝梗のリストによる表記と対応する模式図.

2次枝梗	3次枝梗	穎花, 枝梗の退化	リストによる表記	模式図
なし	なし	なし	[o, o, o, o, o, o, o]	
あり	なし	退化した穎花, 退化した2次枝梗の痕跡あり	[[x, o, o], [], x, o, o, o, o]	
あり	なし	なし	[[o, o, o], [o, o, o], o, o, o, o, o]	
あり	あり	退化した3次枝梗の痕跡あり	[[[o, o], o, o], [], o, o], o, o, o, o, o]	

1次枝梗の基部から先端に向かって記録した

リストでは、退化穎花を“x”，現存穎花を“o”，退化した枝梗を“[]”で示した。模式図では、退化穎花を「×」で、現存穎花を「○」で示した。



第1図 プログラムのフローチャート.

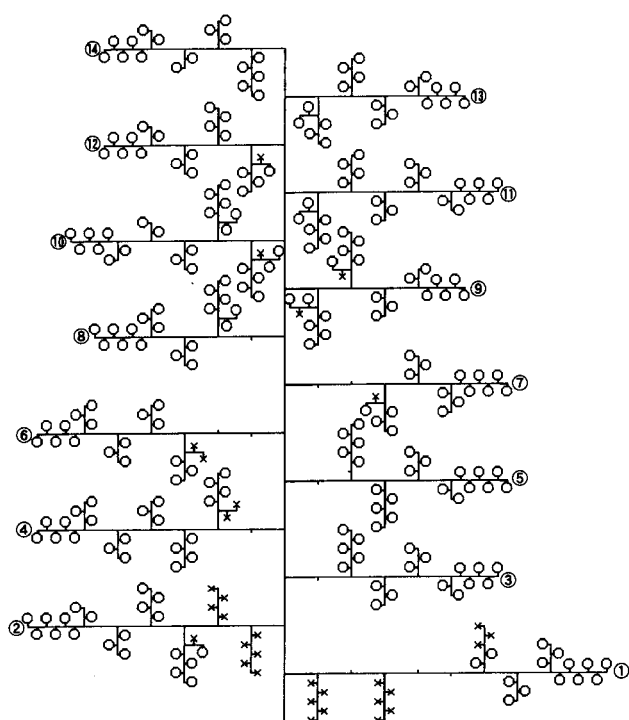
最初に穂全体の分枝構造を記述したリスト形式のデータを入力し、変数“n”に0を代入することにより、1次枝梗を表すリスト形式のデータを処理するときはn=1、2次枝梗を表すリスト形式のデータを処理するときはn=2となる。

分枝の発生や穎花の着生する間隔は、“n”に対応してパラメータとして設定されている。

```

[
①[[x,x,x,x,x],[x,x,x,x,x],[[o,o,x,x,x],[o,o,o],[o,o,o],[o,o,o,o,o],
②[[x,x,x,x,x],[x,x,x,x],[[x,o],[o,o,o,o],[o,o,o,o],[o,o,o],[o,o],[o,o,o,o,o,o],
③[[[o,o,o,o,o],[o,o,o],[o,o,o],[o,o],[o,o,o,o,o],
④[[[x,x],[o,o,o,o],[o,o,o,o],[o,o,o],[o,o,o],[o,o,o],[o,o,o,o,o],
⑤[[[o,o,o,o,o],[o,o,o,o,o],[o,o,o],[o,o],[o,o,o,o,o,o],
⑥[[[x,x],[o,o,o],[o,o,o],[o,o,o],[o,o,o],[o,o,o,o,o],
⑦[[[x,o],[o,o,o],[o,o,o],[o,o,o],[o,o,o,o,o],
⑧[[[o,o],[o,o,o,o],[o,o,o],[o,o,o],[o,o,o,o,o],
⑨[[[o,x,o],[o,o,o,o],[x,o],[o,o,o],[o,o],[o,o],[o,o,o,o,o],
⑩[[[x,o],[o,o,o,o],[o,o],[o,o,o,o],[o,o,o],[o,o],[o,o],[o,o,o,o,o],
⑪[[[o,o],[o,o,o,o],[o,o,o,o],[o,o,o],[o,o],[o,o],[o,o,o,o,o],
⑫[[[x,o],[o,o,o],[o,o,o,o],[o,o,o],[o,o],[o,o],[o,o,o,o,o],
⑬[[[o,o],[o,o,o],[o,o,o,o],[o,o,o],[o,o],[o,o],[o,o,o,o,o],
⑭[[[o,o,o,o],[o,o,o],[o,o],[o,o],[o,o],[o,o,o,o,o]
]

```



第2図 穎花・枝梗の退化を示すリストと描画された模式図（品種：タカナリ）。穂全体および各枝梗のリストは、いずれも基部から先端に向かって記録した。

1次枝梗のリストの直前に、穂首側から数えた順序を説明のために丸数字で示し、模式図中の対応する1次枝梗の先端にも同じ丸数字を示した。

リストでは、退化穎花を“x”，現存穎花を“o”，退化した枝梗を“[]”で示した。模式図では、退化穎花を「x」で、現存穎花を「○」で示した。

3. リスト・データのファイル形式

分枝構造を記録したリスト・データは、プログラムの実行時にキーボードから直接入力することも可能であるが、あらかじめテキスト形式のファイルとして作成・保存しておいて、Prolog システムに読み込ませることも可能である。改行コードは Macintosh 形式の CR のみが望ましいが、DOS 形式の CR+LF でも動作する。

また、ひとつのファイルに複数の穂のリスト・データを

記述することもできる。この場合には、個々の穂を特定するための検索用キーを設定する。たとえば、コシヒカリの穂3本の分枝構造を Koshi 1, Koshi 2, Koshi 3 というリスト・データとしてそれぞれ記述した場合には、1, 2, 3 を検索用キーとして、

p_koshi(1, Koshi 1).

p_koshi(2, Koshi 2).

p_koshi(3, Koshi 3).

という3行からなるファイル（実際には Koshi 1~3 にはそれぞれのリスト・データを入力する）を作成する。リスト・データを入力したり、内容を確認する場合の可読性を高めるため、テキストファイルに記述されているリスト・データを、1次枝梗単位で改行したり、半角のスペースを挿入することも可能である。

4. 分枝構造の模式図の定義と描画プログラムの作成

穂の分枝構造を描画するプログラムは、プログラム言語 LOGO に実装されているタートル・グラフィックス（パート 1982）の一部の機能を Open Prolog に移植し、その上で描画コマンドを処理するという手順で作成した。

描画にあたっては、松葉（1991）の示した穂型と同様に、分枝構造の模式的な図示を目的としたため、分枝の長さや太さ、穎花の形状などは考慮しなかった。また、分枝は主軸に対して 90 度の角度で、左右交互に発生するものと仮定した。現存穎花は「○」、退化穎花は「×」として描画した。枝梗そのものの退化が推定される場合には、ごく短い分枝のみを描画した（第1表）。実際に観察された水稻品種タカナリの穂に着生する枝梗・穎花の退化状況を、リスト形式による記述と模式図を対応させて、第2図に示した。現存穎花 268、退化穎花 32 がそれぞれ着生するこのように複雑な分枝構造も、リスト形式で記述することにより、自動的に描画することが可能である。

なお、Open Prolog はグラフィック画面を印刷する機能を備えていないため、Mac OS の機能により画面をファイルとして保存し、他のプログラムで加工・印刷する必要がある。

5. データ形式の変換や定量的データの表示のためのプログラムの拡張

本プログラムは、穎花の現存・退化という情報以外にも、さまざまな形質に適用可能である。穎花が稔実したか、それとも不稔であったかというような定性的な情報であれば、稔実穎花を“o”、不稔穎花を“x”という要素で記述したリスト形式のデータを作成し、稔実穎花を「○」、不稔穎花を「×」として描画するという簡単な変更で対応できる。

新鮮重や乾物重、窒素含有率、アミロース含有率など、個々の穎果の定量的な情報を分枝構造の模式図に重ねて表示するには、あらかじめ定めた適当な階層区分に従って、稔実穎花を示す「○」の内部を塗り分ける必要がある。この場合には、測定値をリストの要素として直接入力する。ただし、Open Prolog は整数しか処理できないので、測定値の有効桁数に応じて 10 のべき乗を乗じて整数化する必要がある。それ以外は、定性的な情報の場合と同様に、テキスト形式のファイルとして保存すればよい。

ただし、“o”や“x”などの文字を用いて簡潔に入力できる定性的な情報とは異なり、定量的な情報は桁数が多

第2表 表形式による穎果の新鮮重の記録例。

番号	1次枝梗の位置 (i)	1次枝梗内の位置 (j)	2次枝梗内の位置 (k)	新鮮重 (mg)
1	1	1		19.02
2	1	2		0.00
3	1	3		19.76
4	1	4		18.62
5	1	5		18.92
6	1	6		19.45
7	2	1	1	11.96
8	2	1	2	0.00
9	2	2	1	16.22
10	2	2	2	15.29
11	2	2	3	12.30
12	2	3		17.36
13	2	4		19.30
14	2	5		20.17
15	2	6		21.59
16	2	7		20.44
17	2	8		19.76
中略				
81	9	1		20.91
82	9	2		19.45
83	9	3		20.44
84	9	4		21.24
85	9	5		20.08
86	9	6		20.59

品種コシヒカリの1次枝梗の位置 (i)、1次枝梗内の位置 (j) および2次枝梗内の位置 (k) を、いずれも基部側から数えた。

不稔穎花は、新鮮重 0.00 mg として示した。

く、リスト・データを直接作成することが煩雑となる場合も想定される。鈴木ら（1994）はコシヒカリの新鮮重を第2表のような書式で記録している。この表において、穎花の着生位置は、

穂軸の基部から i 番目の1次枝梗の、

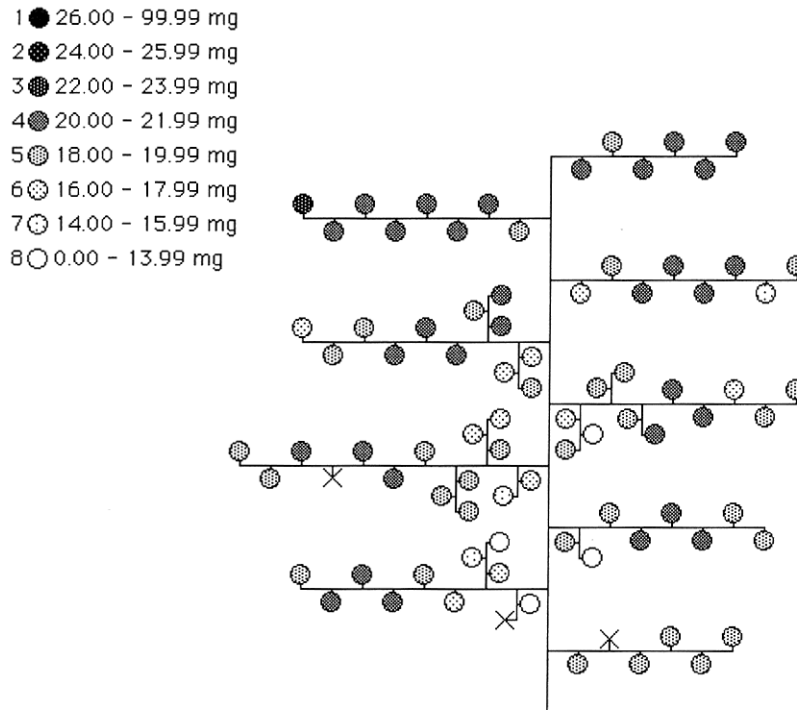
基部（穂軸）側から j 番目に着生する穎花、あるいは

基部（穂軸）側から j 番目に着生する2次枝梗の基部（1次枝梗）側から k 番目に着生する穎花、

という表形式で特定され、測定値が記録されている。そこで、表計算ソフトウェアやエディタを用いてこの形式で作成され、空白を区切り文字としたテキスト形式で保存されたデータを、Prolog のパターン照合機能を活用してリスト形式のデータに変換するプログラムを作成した。この変換プログラムにより、定量的データを要素とするリスト・データが出力され、第2表のデータを第3図のように塗り分けることが可能となった。さらに、第2表とは記録の様式が異なる場合も、Prolog による変換プログラムの作成は比較的容易であると推察された。

6. プログラムの配布

本プログラムは、独立行政法人農業技術研究機構職務作成プログラム規定に基づく申請を行っている。利用を希望する場合は、中央農業総合研究センター企画調整部連絡調整室運営係（電話 029-838-8562）に連絡し、所要の審査



第3図 穎果の新鮮重と穂上着生位置との関係（品種：コシヒカリ）。
 不稔穎花は「×」で示し、稔実穎果は「○」の内部を新鮮重の階層に応じて塗り分けた。

A {
 reverse([], []).
 reverse([A|L], L3) :- reverse(L, L2), append(L2, [A], L3).
 append([], L, L).
 append([A|L], L2, [A|L4]) :- append(L, L2, L4).
 B {
 p_ord([], L, L).
 p_ord([A|B], Ord, L) :- reverse(A, C), append([C], L, L2), p_ord(B, Ord, L2).
 C {
 nth_sum_up(N, Sum, Count, L) :-
 p_ord(L, Lrev, []), nth_sum_up(N, Sum, 0, Count, 0, Lrev).
 nth_sum_up(N, Sum, Sum, Count, Count, []).
 nth_sum_up(N, Sum, Sum1, Count, Count1, [A|B]) :-
 nth(N, X, Found, A), Sum2 is Sum1+X, Count2 is Count1+Found,
 nth_sum_up(N, Sum, Sum2, Count, Count2, B).
 nth(N, 0, 0, []).
 nth(1, 0, 0, [X|_]) :- not(integer(X)).
 nth(1, X, 1, [X|_]).
 nth(N, X, Found, [A|B]) :- N1 is N-1, nth(N1, X, Found, B).

第4図 1次枝梗先端からn番目の穎花の定量的データを集計するプログラムリスト。

を経て、利用許諾を受けることとなる。

考 察

本報告では、イネの穂の分枝構造あるいは着粒構造を、分枝を単位としたリスト形式で表記する方法を提案するとともに、その分枝構造を平面上に模式図として描画するためのコンピュータプログラムと、さらに一粒単位で得られた定量的データをその模式図に重ねて表示する方法を示し

た。その結果、一見複雑に見える穂の着粒構造は、リスト形式の記述により単純で、かつすべての分枝に共通の構造の繰り返しであることが示され、描画された模式図は直観的な理解を助ける上で有用であると考えられた。そこで、リスト形式で記述された定量的データの解析や、本プログラムのより広範な応用について述べる。

本プログラムの作成に採用した Prolog は、プログラムの実行を Prolog システムそのものが制御する。したがっ

て、繰り返しや条件分岐などの制御命令をプログラム中に記述する必要がなく、簡潔なプログラムが作成できるという利点を有する。その一方で、プログラムの実行時には逐次的に翻訳するインタプリタ型の処理を行っているため、他の言語で作成した同様の処理を行うプログラムと比較すると実行速度が遅い欠点を有する。しかし、現在ではコンピュータの処理速度が飛躍的に向上したため、インタプリタ型処理のデメリットも相対的に小さなものとなっている。実際、Mac OS 9.2.2 環境下で、400 MHz で動作する PowerPC G3 プロセッサと 256 MB のメモリーを搭載した iMac DV Special Edition で実行した場合、本報告において作成したプログラムの実行時間はいずれも短く、第2図に示したタカナリのような複雑な構造でも、ようやく画面上での描画の進行が確認できるほどであり、第3図の塗り分けは瞬時に終了した。したがって、処理速度には実用上の問題はない。

上述したように、本プログラムで活用した再帰的呼び出しは、要素の中にリスト・データが含まれ、要素の個数が不定である穂の分枝構造を記述したリスト・データの処理にきわめて有効な手法である。注意すべき点としては、実行時にメモリーを大量に使用する点がある。そのため、1次枝梗数が多い穂や、3次枝梗まで分化している穂の模式図を描画する場合には、Open Prolog システムの使用するメモリーが、推奨値の 4 MB では不足するため、40 MB を割り当てることが望ましい。

本プログラムのリスト・データは、穂軸および分枝の基部側から順に記録しているが、穂軸や分枝の先端側からの着生位置が重要となる場合がある。この場合、Prolog はリスト処理に優れているため、第4図Aに示した4行のプログラムで、任意の長さのリストを逆順に並び替え、先端から記録したリストに変換できる。また、第4図Bの2行を追加すると、穂全体の分枝構造を基部側から記述したようなリスト・データも先端側からの記述に変換することが可能である。さらに、第4図Cの部分を加えることにより、1次枝梗上の先端から数えた任意の位置に直接着生する穎果の定量的データを穂全体で集計し、合計値とデータの個数を出力するプログラムとなる。分枝構造を記述したリスト・データを L とすると、先端から N 番目の穎果を集計するには、

$\text{nth_sum_up}(N, \text{Sum}, \text{Count}, L).$

と入力する。第2表の測定値では、 $N=1$ のとき $\text{Sum}=173.73 \text{ mg}$ 、 $\text{Count}=9$ という結果が得られる。このように、明確に定義されたリスト形式による分枝構造の記述と、柔軟なプログラム言語 Prolog の採用により、今後新たな研究上の必要が生じた場合にも、それに対応して本プログラムを拡張することは比較的容易である。

植物をコンピュータ画面上に一定の規則に基づいて表示する手法は、デュードニー (1989) により紹介されている。本プログラムにより描画された模式図は、枝梗・穎花

の着生順序をリスト形式で記述したデータから作成されている。個々の穎花の位置関係は正しく表現されているが、枝梗の長さ、太さ、あるいは穎花の形状などの情報はリストには反映されていないため、元となった穂と模式図とは相似ではない。そのため、位置関係よりも長さなどのサイズがより重要な場合には、実際の植物の各器官を立体として計測し、コンピュータ上に3次元画像として再構築する手法がより適している (Watanabe ら 1999)。しかし、個々の穎花の持つ情報と着生位置との関係の図示においては、精緻なシミュレーション画像よりも、簡易な模式的平面画像による視覚化の方がより理解の助けになると考えられる。第2図に示すように、基部から2本目までの1次枝梗では2次枝梗に着生した穎花の退化が顕著であること、先端から6本目までの1次枝梗には2次枝梗そのものの退化は認められなかったことが、それぞれ容易に把握できる。また、第3図からは、2次枝梗に着生した穎果の新鮮重が1次枝梗に着生した穎果より小さい傾向があることが、一目で理解できる。

以上のように、この描画プログラムは、イネの穂を模式的に図示するために作成したものであるが、分枝的な構造を示すさまざまな対象について応用が可能である。廣井ら (1990) の示したイネ属における多様な穂の構造も、本プログラムを用いた模式化により直観的な理解が可能となると推察される。また、180度の間隔で交互に発生 (互生) するイネの分げつ体系の表示にも、プログラムの大幅な変更なしに適用できると考えられる。さらに、分枝が左右両方に発生する対生や、分枝する角度が90度より小さくなる輪生など、植物の多様な分枝構造の描画にも適用可能と考えられる。

引用文献

- 荒井邦夫・河野恭広 1978. 水稻の穂の発育に関する研究. 第1報 穂上位置別にみた穎花の発達の特徴. 日作紀 47: 699-706.
- 荒井邦夫・河野恭広 1979. 水稻の穂の発育に関する研究. 第2報 穂上位置別穎花の窒素集積パターンに及ぼす出穂期追肥の影響. 日作紀 48: 335-342.
- デュードニー, A.K., 1989. コンピューターレクリエーション II. 遊びの探索. 山崎秀記監訳. 日経サイエンス社, 東京. 115-120.
- フィリピン, B. 1990. Prolog ユーティリティライブラリ. 中島誠・伊藤哲郎訳. 海文堂, 東京. 1-149.
- 福嶋陽 1999a. イネの1穂穎花数を規定する穂の分枝構造に関する発育形態学的解析. 日作紀 68: 71-76.
- 福嶋陽 1999b. イネの1穂穎花数を規定する穂の分枝の分化・発育に関する発育形態学的解析. 日作紀 68: 77-82.
- 廣井清貞・A.A. Mamun・和田富吉・武岡洋治 1990. イネ属植物における穂構造の種間変異に関する研究. 日作紀 59: 815-823.
- 堀内久満. 1991. 福井県における水稻良食味品種育成の現状と問題点. 育種学最近の進歩 33: 16-23.
- 石川哲也・藤本寛・丸山幸夫 1998. 多収性水稻品種タカナリの分げつ特性および穎花形成特性. 日作紀 67(別1): 14-15.

- 石川哲也・藤本寛・柊木信幸・丸山幸夫・秋田重誠 1999. 水稻品種タカナリの出穂期までの乾物生産と穎花数の決定. 日作紀 68: 63—70.
- 木戸三夫・梁取昭三 1968. 腹白, 基白, 心白状乳白, 乳白米の穂上における着粒位置と不透明部のかたちに関する研究. 日作紀 37: 534—538.
- 黒川利明 1985. Prolog のソフトウェア作法. 岩波書店, 東京. 57—113.
- 松葉捷也 1991. イネの穂の着粒構造の分析およびその形成機構論. 中国農研報 9: 11—58.
- Matsue, Y., K. Odahara and M. Hiramatsu 1994. Differences in protein content, amylose content and palatability in relation to location of grains within rice panicle. Jpn. J. Crop Sci. 63: 271—277.
- パパート, S. 1982. マインドストーム. 子供, コンピューター, そして強力なアイデア. 奥村貴世子訳. 未来社, 東京. 69—110.
- 進藤久美子・豊島英親・安井明美 1998. 玄米 1 粒中無機元素含有量の穂上位置による差異. 日作紀 67: 478—484.
- 鈴木保宏・石川哲也・池上勝・建部雅子・小前幸三 1994. イネの 1 穂内の玄米タンパク質含量の着粒部位別の変異. 育雑 44(別 2): 251.
- 田代亨・江幡守衛 1974. 腹白米に関する研究. 第 2 報 穂上位置と腹白米の発現. 日作紀 43: 105—110.
- Watanabe, T., J. H. Hanan and P. M. Room 1999. Virtual Rice: I. Measurement and specification of three-dimensional structure. Jpn. J. Crop Sci. 68(Ext. issue 2): 68—69.

A Computer Program Written in Prolog to Illustrate the Branching Structure of Panicles of Rice (*Oryza sativa* L.): Tetsuya ISHIKAWA^{*,1)} and Yasuhiro SUZUKI²⁾ (¹⁾Natl. Agr. Res. Cent., Tsukuba, 305-8666, Japan; ²⁾Natl. Inst. Crop Sci.)

Abstract: It is possible to analyze the chemical constituents or physical properties of each rice grain, and attempts are being made to clarify the relationship between the analytical results and position of the grain on the panicle. The objective of this study is to develop a computer program for schematic presentation of the branching structure of rice panicle, and for illustrating the information of each grain corresponding to their position on the panicle. The branching structure of panicle was described in list-format in order to deal with their diverse structures. A programming language Prolog, with unique characteristics such as pattern matching and unification, was used to develop the drawing program by fully utilizing its recursive call function. For the effective utilization of the analytical results already recorded in table-format, another Prolog program was developed to convert them into the data described in list-format. Besides illustrating the qualitative information, such as existence or degeneration of spikelet, another function to illustrate the grains in specific colors, according to their quantitative information, such as fresh weight, was added to the drawing program. Although the drawn schematic illustration was not accurate in terms of the length of rachis and shape of grain, the relative positions of rachises and grains were precisely expressed, and it was concluded that this illustration was helpful in understanding the relationship between the qualitative or quantitative properties of grains and their positions on the panicle intuitively. Application of this drawing program to the panicle of Genus *Oryza*, which has similar branching structure, or the tillering system of rice, is thought to be possible.

Key words: Branching structure, List, Panicle, Prolog, Recursive call, Rice.