□　データベース関連

①　データベースの基本

②　データベースの内容／形態による分類

・文献データベース／ファクトデータベース

これらの呼称はデータベース内のデータの内容に従ったものである。　データベースの内容が文献に関するものである時、文献データベースと称される。　また、データベース中に保存されているデータの種類がファクトデータ（物性データ、薬理データ、スペクトルデータ等）である時、このデータベースはファクトデータベースと呼ばれる。　化学分野ではこのファクトデータベースが多数存在する。

以下に化学やバイオテクノロジー関連分野で利用されているデータベースの内容を列挙する。

文献／特許／数値データ（種々物性）／スペクトル／薬理データ／X線結晶解析／プロテイン／ＤＮＡ／その他

・パブリックデータベース／インハウスデータベース

データベース自体が一般に公開されているのか、されていないかで2種類のデータベースに分類される。　一般に公開されるものをパブリックデータベース、公開されずに企業内、あるいはグループ内で専用に利用されるものをインハウスデータベースという。

③　データベースモデルの種類と概要

データベースはそのデータを管理する方法で様々なモデルがあり、この管理（データ間のネットワーク化）方法の違いにより幾つかのモデルに分類する事が可能である。　実用化されている典型的なモデルとしては3種類ある。

> 階層型モデル
> ネットワーク型モデル
> リレーショナル（関係）型モデル

これら3種類のモデルのうち階層型及びネットワーク型モデルはデータを互いにネットワーク化するもので歴史が古く、実用システムとして大規模なものが構築されている。最後のリレーショナルモデルの歴史は3モデルの中では最も新しいものである。　このモデルは化学（自然科学一般ともいえる）分野では文献検索等を除けば、物性／毒性／スペクトル等様々なデータを扱うデータベースとして最も利用されているモデルである。　このリレーショナルモデルの基本概念（データ構造）は、科学者が日常的に用いているデータ整理手法とさほどかわりがなく、科学者には受け入れやすいモデルである。

以下前記3種類のデータモデルについて順にのべる。

□　階層型モデル及びネットワーク型モデル

階層型モデル及びネットワーク型モデルはいずれもデータ同志の連携をさせる為にネットワークを形成させるもので、2モデルの差はネットワークの形式の差による。

・階層型モデル

階層型モデルではデータの相互関係が階層的になり、且つデータ間の横のつながりが比較的弱いようなデータ構造を持つものに的したデータベースである。　このモデルのデータ間のつながりの形は基本的に木（ＴＲＥＥ）構造をなしている。　この木構造にも様々

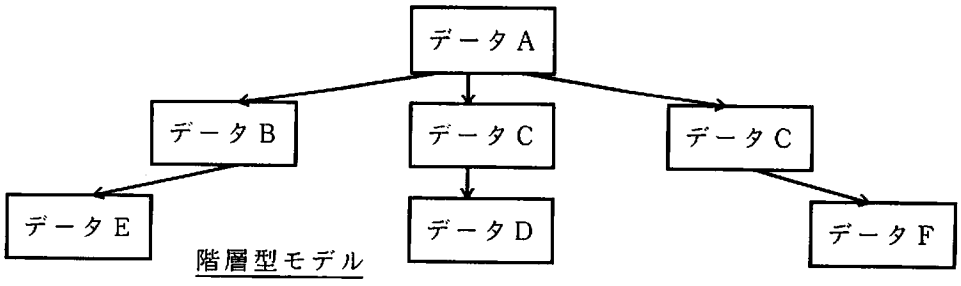


図　．階層型モデルのデータ構造

なバリエーションが存在し、個々のデータベースに適した形態をとるようにする事が必要である。
階層構造を持つデータは一般的な分野では多数存在する。 最も典型的な例としては家系図がある。 また、さまざまな構造体とその部品との関係もツリーとして構成することが可能である。

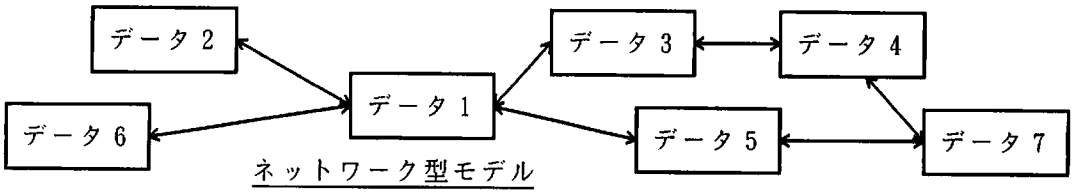
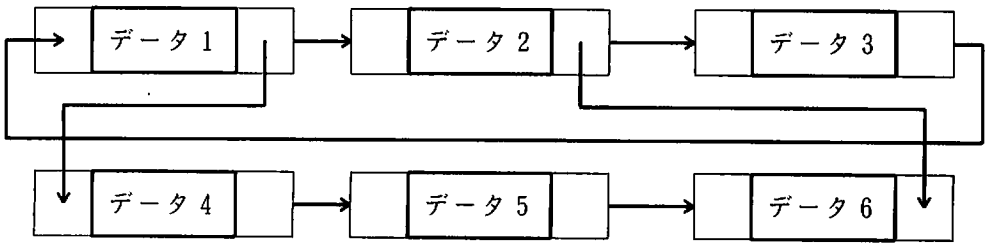


図 . ネットワーク型モデルのデータ構造

ネットワーク型モデルはより複雑な相互関係を持つデータを扱う時に利用されるモデルである。 個々のデータ間に高度な階層関係や相互関係が存在し、これらが互いに交錯しているようなデータを扱う時に利用される。

□ 多重つなぎ編成によるデータベース構築

この多重つなぎ編成はネットワーク型モデルの一種である。 このアプローチではデータ間のネットワーク関係（数学的にはリスト：ＬＩＳＴと呼ばれ、データベース関連では鎖：ＣＨＡＩＮと呼ばれる）をそのままデータベース構造に反映させてデータベースを構築するものである。 従ってこのデータベースは高速化を狙うよりも、データ間の結合関係を重視したデータ検索を目的としている。
データは本来のデータ自身に対する情報と、そのデータと関連するデータへの結合情報部分とから構成される。



＊太線枠内は保存用データ、細線枠内が結合情報を持っている部分

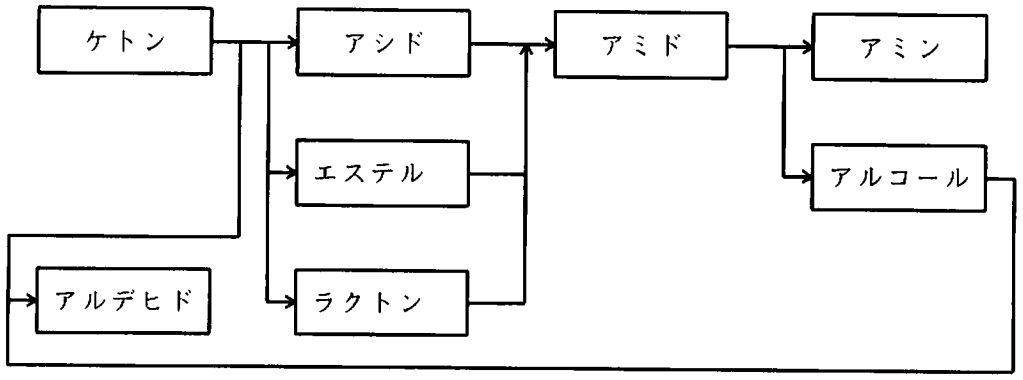
図 . 多重つなぎ編成によるデータ連携関係例

図 からもわかるがこの多重つなぎ編成によるデータベースではデータ間の“リンケージ情報”が情報として大きな部分を占め、この情報がデータ間の複雑な相互関係の反映に役立っている。 従って、データ間に複雑な階層関係や結合関係がある時にはこの方法がとられる事が多い。 この多重つなぎ編成の特徴を以下にまとめる。

データが相互にリンクされているので、この結合関係を用いて連続的なデータの呼出が可能である
複雑な関係を有する情報も取扱可能である

このアプローチの欠点としては、結合関係を重視している為にデータの独立性が相対的に低くなる事。 また、データの結合関係が複雑になるとデータベースの操作やメンテナンスが複雑になる事などである。

例）ケトンを基点とした官能基群を考慮し、化合物データベースを構築する



□ リレーショナルモデル（関係モデル：RELATIONAL MODEL）

リレーショナルモデルは自然科学分野で頻繁に採用されている。 これは、リレーショナルモデルで扱うデータ形式が自然化学分野におけるデータ構造の表現に都合のよい形式であることが大きな理由である。 すなわち、リレーショナルモデルは1サンプル対多項目というマトリクスを基本としたデータ構造を持つものを扱うのに適している。 例えば一つの化合物（サンプル）に関する様々な物性（項目）といった関係を持つデータであり、このような関係を持つデータは化学に限らず自然化学分野では頻繁に現れる。

このようなデータは、先にのべた階層型やネットワーク型のモデルにしてデータベース化することは困難である。 むしろ研究者が昔から行っていたデータの管理形式、すなわちサンプルを行とし、項目を列としたマトリクスで容易に管理する事が可能である。 リレーショナルモデルはこのマトリクスを扱うためのデータベースといえる。

このリレーショナルモデルは以下に示す関係式で表すことが可能である。

$$R(X_1, X_2, \cdots\cdots X_n) = \{(x_1, x_2, \cdots\cdots x_n) \mid x_i \in X_i \wedge \text{命題}R(x_1, x_2, \cdots\cdots x_n)\text{が真}\} \subseteq X_1 \times X_2 \cdots X_n$$

X_i は関係の定義域（DOMAIN）、定義域の数を関係の度数（DEGREE）、
x_i は関係の要素、$(x_1, x_2, \cdots\cdots x_n)$ はn組（n－TUPLE）と呼ぶ。

第1マトリクス

関係名	定義域 1	2	3	4	5
実験者	実験者名	実験番号	実験日付	天気	実験結果
試薬	試薬名	購入日付	製造メーカ	残量	
化合物	化合物ID	製造者名	製造日付	機器分析	薬理データ
……	……	……	……	……	……

第2マトリクス

化合物	1 化合物ID	2 製造者名	3 製造日付	4 機器分析	5 薬理データ
化合物1	621A	KY	3.01.22	IR, NMR	11C
化合物2	372X	KK	3.02.15	IR, NMR	11C
化合物3	277C	TO	3.02.05	IR, MASS	12C
……	……	……	……	……	……

前例ではマトリクスが2階層になって管理されている。 第1マトリクスは種々のデータの関係に関する情報を一つのマトリクスとした管理用（またはインデックス用）のマトリクスである。 第2マトリクスは第1マトリクスにより定義された個々の関係毎に設けられ、これら関係に関する実際のデータが入っているマトリクスである。

□　スプレッドシート型モデル

最近パソコンやワークステーション上で頻繁に利用されるデータ管理形式として、スプレッドシートというものが利用されている。　このスプレッドシートは名前のとおり、概念的にはデータが書かれた一枚のシートをあるまとまったデータ単位として扱うものである。　前記リレーショナルデータベースを基本として考えるならば、一つのスプレッドシートは一つのマトリクスに該当することになる。

このマトリクス１個を一つのシートとみなし、一枚あるいは複数枚のシートを用いて計算機上で利用される。　このスプレッドシートはＬＯＴＵＳやＥＸＣＥＬ等の表管理プログラムの基本となっている。

化学の分野でもこのスプレッドシートが利用されはじめている。　他の分野のスプレッドシートとの違いの大きな点は、第１列目の項目として化合物ＩＤや化合物の構造式を用いることである。　それ以外の列に個々の化合物に関する情報が配置されている。
このような形式のスプレッドシートを特に“ケミカルスプレッドシート”と呼んでいる。
ケミカルスプレッドシートの一例を表　に示す。

表　．ケミカルスプレッドシート例

化合物構造式	融点	沸点	薬理データ	LOGP	……
化合物　1	11.4	120.8	134	4.3	……
化合物　2	-5.8	186.4	155	2.7	……
化合物　3	123.3	256.1	176	3.3	……
……	……				
……	……				

□　オブジェクト思考型モデル

④　実際のデータベース構築時に利用される様々な技術

データベースの構築では検索対象となるデータの量がかなり大きなものとなる為、いくつかの点で注意が必要である。

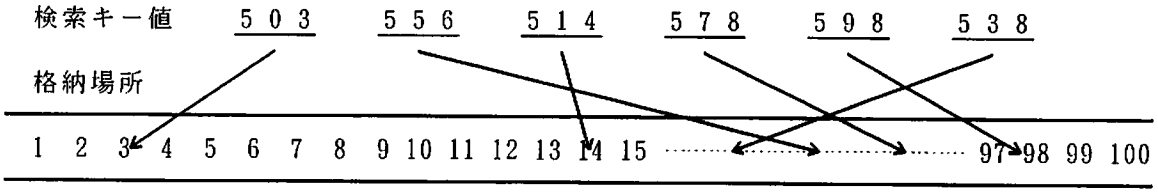
1．検索スピードがデータベース中のデータ量と関係なく一定で速い事。
2．データの量が多い事から記憶領域の効率的な利用が必要。
3．データベース内のデータの特性を反映する構造である事。

これらの条件を満たす為の技術としていくつかの技術が確立されている。　ここではこのような技術の中で計算機上での一般的な技術について簡単にまとめる。

□　ＨＡＳＨＩＮＧ　ＦＵＮＣＴＩＯＮ（散らし関数）による高速化

散らし関数により検索キーの数値データを変換し、その変換された値をデータベース上の格納場所、あるいは被検索体のＩＤ番号等にする手法である。

いま格納場所／ＩＤを示す値が１～１００の整数で示されているとして、被検索体の検索キーの値が５００～６００の値であるならば、この値を１００に近い素数９７で割り、残った余りの値をデータが格納されている場所の番号やＩＤに対応させることで直接被検体を捜し出す事が出来る。（割り算法）



つまり全く関係の無い２つの数値をハッシング関数を用いる事で関係付け、１対１の高速検索を実現させるものである。　この関係は検索用の数値データをｘ、被検索体が収容されている場所を示す値をｎ、ハッシング関数をＨとすると以下の式で示される。

$$n = H(x) \qquad (\ )$$

この検索の特徴は検索キーの値による１体１の高速検索が可能となる事である。　このハッシング（散らし）により、被検体が格納場所上に均等に分散されているならば、データベースの大きさと関係無く常に一定の速さで検索する事が可能となる。

ハッシングの特徴を生かした高速検索を保証する為にはこのハッシング（散らし）がうまく行われるか否か（データ格納アドレスがメモリー上で均等に割り振られるか否か）に依存する。　一般的にこのハッシングをうまく行う為には以下に示す２つの要素を検討する事が必要である。

①　実際のデータ（被検索体）の数よりも大きな格納場所を用意する事が必要

データベースの大きさが大きくなるとハッシング関数を用いても同じ番号に被検索体が複数アサインされる事が生じてくる。　この同義語（ＳＹＮＯＮＹＭ）の発生が多くなると検索効率が著しくダウンする。　この同義語の発生を少なくするにはハッシング関数を再考する事と、被検体の格納領域を大きくする事が考えられる。　この格納領域は計算機の主記憶の容量や記憶領域の有効利用等からベストと思われる大きさを判断する事が必要である。（一般的にはデータ数の２～５割増しといわれている）

②　数値データの内容に応じた適切な散らし関数を採用する事が必要

散らし関数の良否が検索効率を大きく左右する。　散らしの目的は被検索体を格納場所に均一に割り当てる事にある。　この散らしが均一に行われないと、一つの場所に複数の被検索体がアサインされたり、かたよってアサインされる事になり検索スピードのダウンやメモリー利用効率の低下等の問題が発生してくる。

この散らし関数もさまざまなものがある。　最初の例に示した割り算法の他、混合法（キー値ｘ定数Ａ／定数Ｂの余りを用いる）、中央二乗法（キー値を二乗し、中央部分のある一定の桁数を取り出し、その値を用いる）等目的に合わせて選択する事が必要である。

例え理想的な設計をしたとしても同義語の発生を０とする事は困難である。　この同義

語が発生した時の回避方法として幾つか実用化されている手法がある。

- 一つの番地に複数の被検体の番号が格納できるようにする
- 同義語が発生した番地の前後を開放し、そこに同義語の被検体を収納する
- 同義語だけをまとめて収納する場所を予め設定しておく

（このような同義語の格納場所はバケツ（ＢＵＣＫＥＴ）と呼ばれる）

これらのアプローチのどれを採用するかは、メモリーの容量や検索スピードの問題等の関係で、個々のシステム単位で最適のものを採用する事が必要である。

利用例）

化合物の原子量をキー値として化合物（４０００個）の検索をする。

化合物の分子量Ｍは２００～６００の間の値を取り、化合物の格納場所Ｄは１～５０００（２５％増）とする。　この時用いる散らし関数としては以下のようなものを用いる。

$$D = int\left(\left(\left(\frac{M}{5001}\right) - 0.03997\right) \times \frac{5}{8} \times 10^{5}\right) \quad \text{（　）}$$

上式でｉｎｔは切り捨てを意味し、基本的な手法は割り算法を採用するとする。　従って、分子量を格納場所の数５０００に最も近い素数５００１で割り、値の基準を合わせる為にその値から０．０３９９８を引く。　この値を格納場所の１～５０００にあわせる為に５００００／８をかけ、この値を切り捨てることで格納場所のＩＤ番号を得る。

実際の手続きとしては効率良い検索を実現する為にもデータの格納場所は、被検索体４０００の２割から５割程度（ここでは格納場所として５０００を用いた）大きめにするのが良い。　さらに、散らし関数を用いても同義語（同一アドレス）をさける事は困難であること。　また、今回用いた分子量は分子式が同じならば異なった化合物であっても同じ分子量となり、全く同一の検索キーを持ち同義語の発生原因となる。　これらの点からも同義語が少なからず発生する事は明白である。　このような同義語を取扱う為、同義語の格納領域を予め設定しておくことが望ましい。

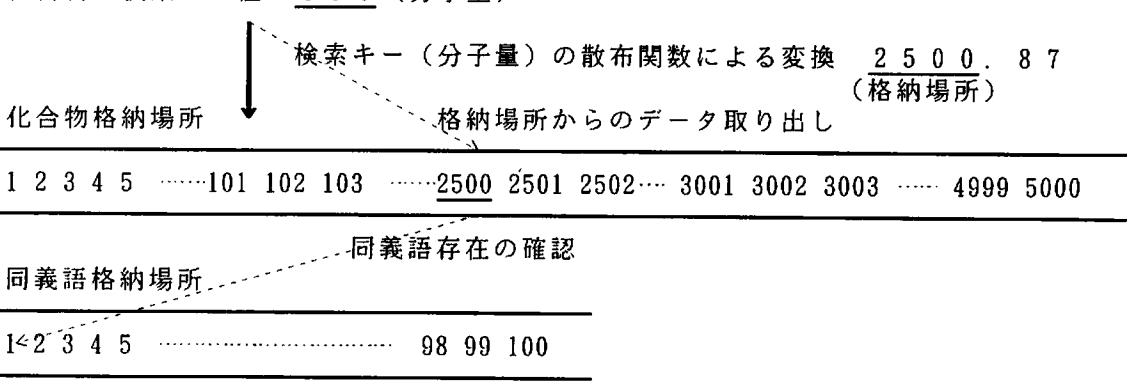


図　．ハッシング関数を用いた化合物の保存形式

□　インバーテッドファイル（ＩＮＶＥＲＴＥＤ　ＦＩＬＥ：（転置ファイル））の作成による高速化

文字どおりファイルの中身（行列）を反転させる手法を意味している。　形式上は行列を反転するのであるが、各行あるいは列単位にある特定の条件を満足するデータ群を取り出して新たなファイルをつくる。　このファイルの利用目的は検索時のインデックスに利用するというものである。　行を単位にするのか列を単位として展開するかは個々の目的に応じて採用される。

一般的にデータはマトリクス（行列）の形式で表現される。　検索という観点から考える時、行の要素としては検索目的となる項目を取り、列の要素としては検索を行うに必要となるデータ項目を指定しておくことが多い。

被検体ＩＤ	検索項目 1	2	3	4	……	n－1	n
1	0	0	1	1	……	0	1
2	1	0	1	0	……	1	1
3	0	1	0	1	……	1	0
……	……	……	……	……	……	……	……
……	……	……	……	……	……	……	……
N	1	0	1	0	……	0	0

↓ インバーテッドファイル化（データ中１のものだけ抜き出してくる）

検索項目	被検索ＩＤ 1	2	3	4	……	D－1	D
1	2	5	10	12	……		
2	3	4	7	18	……		
3	1	2	8	11	……		
……	……	……	……	……	……	……	……
……	……	……	……	……	……	……	……
n	1	2	6	21	……		

＊Ｎ：被検索対象サンプル数（行）
ｎ：検索項目数（列）
Ｄ：インバーテッドファイルで新たに取り出されたサンプルの数
総てのサンプルＮがインバーテッドファイル中に採用された時Ｎ＝Ｄとなる
通常はＮ＞Ｄである

このようなマトリクスを利用する時、計算機のファイルにはそのままの形式でストアさせるだけでなく、個々の項目の高速検索を目的として特別のファイルを設ける事がある。これがインバーテッドファイル（転置ファイル）と呼ばれるものである。

このインバーテッドファイルは基本となるファイルの行と列とを入れ換え、新たに検索用のファイルとして作成されたものをいう。　通常このファイルは列となっている項目（例えば融点／沸点／分子屈折率等）毎に設けられる。　従って、項目数がｎ個の時、インバーテッドファイルの数は最大ｎ個となる。

このインバーテッドファイルを用いない時、検索項目のパターンに従って検索するならば検索は総ての被検体サンプル（Ｎ）についてチェックを行う事が必要になり、被検体の数に比例して検索時間が増大してゆく。　しかし予め検索項目を行に設定し、その検索項目を満たすパターンについてのＩＤ番号を列に設定したインバーテッドファイルを用意しておけば検索スピードが向上し、検索上での様々な操作（ＡＮＤ／ＯＲ検索等）を簡単に実行する事が可能となる等の利点が生じる。

例えばある一つの項目で検索を行いたい時、このインバーテッドファイルがなければこの検索項目に従って対象となる総ての被検体をチェックすることが必要である。　計算機ではこのようなデータ検索はディスクへのアクセスを要求する事がおおく、且つこのディスクアクセスは計算機泣かせの最も時間のかかる作業である。　データベースの設計ではこのディスクアクセスをいかにして少なくさせるかという事がデータベース成否の鍵をにぎっている。　先のインバーテッドファイルに関するアクセスを考えた時、検索キーに従って被検体が分類わけされている為、検索項目が一つであるならばその項目に該当する行の部分に存在する被検体を取り出す事で検索は完了する。　この検索項目が複数ある時はインバーテッドファイルが無い時と比べるとさらに効果が出てくる。

このインバーテッドファイル作成による検索時の効果をまとめると以下のようになる。

検索スピードが速くなる
検索に要する時間は被検索体の数に影響されることが少なく、ほぼ一定である
複数項目を組み合わせた複合検索が簡単で、スピードも速い

注意すべき点としては本来のデータ格納領域の他にインバーテッドファイル用の格納領

域も必要となりメモリーを多く必要とすること、及びデータベースの更新が発生するとインバーテッドファイルの更新も必要になるという事である。　従って、頻繁にデータの更新が発生するデータベースではメンテナンスという観点からはインバーテッドファイルの長所をいかし切れなくなる事がある。

＊インバーテッドファイルは大きなデータベースとなると項目毎に設ける事も多い。　項目毎にインバーテッドファイルを用意し、且つ項目の内容を細かに分類する事でより細かな検索を高速に達成する事が可能となる。

例）ケトンの有無に関するインバーテッドファイル
ケトンの有無だけの検索ではこのファイルの総ての被検索体を出力する事が必要であるが、このケトンを更に細かく分類する（ケトン、アルデヒド、エステル、アミド、ラクトン等）事でより細かな情報をより高速で検索する事が可能となる。

⑤　日本及び外国のデータベース

・化学文献データベース（ＣＡＳ，ＤＡＲＣ）
・化合物物性データベース（ＣＩＳ）
・化合物スペクトルデータベース（Ｔｏｏｌ－ＩＲ，ＳＤＢＳ）
・Ｘ線結晶データベース
・バイオ／プロテイン関連データベース
・総合データベース（ＤＩＡＬＯＧ等）

⑥　日本の団体

・科情協
・ＪＩＣＳＴ
・ＢＩＤＥＣ

⑦　データベース関連テーブル

第３章　化合物の検索（化合物データベース）

１．化合物検索概論

化合物の検索技術は計算機化学における基本である。　化合物の検索技術は計算機のハード関連技術の向上と共に変化して来ている。
化合物の検索では以下の手続きをとることが必要となる。
①　化合物構造式（検索対象化合物）の計算機への入力
②　データベース中の化合物から化合物を取り出してくる
この２段階の手続きの中に様々な技術が必要となるが、これらの技術はその基本を計算機におくのか化学におくかで２種類に分類できる。　これら様々な技術を駆使して化合物の検索が行われることになる。
計算機に基本をおく技術としては例えば“インバーテッドファイルの利用”ヤ“ハッシング（散らし）関数の利用”等がある。　その他、ファイルのストア形式として階層型データベースやリレーショナルデータベース等様々な形式のものがある。　これらの技術に関しては他の計算機に関する著書に詳しいのでここでは取り上げない。　化学に近い、あるいは化学と計算機を結びつける為の技術を中心としてのべる。

□　検索技術の歴史的変遷

当初計算機のハードが貧弱であった時には、計算速度も遅く、しかもメモリの単価が高いため、このメモリを節約しながらデータベースを作成すること及び検索を高速に行うための技術が最優先技術として要求された。
化合物検索の初期では、化合物の構造式を１次元の文字及び数値からなる文字コードへと変換し、そのコードを検索する事が一般的に行われていた。　化合物構造式の１次元文字列へのコード化により、メモリを節約した化合物データベースの実現や検索が可能となる。　さらに、この文字コードを用いた化合物検索には計算機分野で一般的に利用されてきた文字列検索技術をそのまま適用することが可能となる。　このため、化合物を対象とした新たな検索技術を開発する必要もないので初期の化合物検索システムにはよく利用されたアプローチである。
その後計算機のハードの向上と共にメモリー単価も低くなり、メモリーを充分に用いた化合物表現（結合表等）及び検索技術が幅広く展開されるようになった。　またグラフィックディスプレイの充実とともに化合物構造式のコード化を人間が行う必要はなくなり、化合物構造式をディスプレイ上に直接書き込む事で、あとは計算機が自動的にコード化する事が一般的アプローチとしてとられるようになった。　この結果、当初計算機と化学者との間に存在した大きなコミュニケーションギャップは大幅なＭＭＩ（マンマシンインタフェース）の向上とともに可能な限り小さなものとなりつつある。
以上の特徴を簡単に表１にまとめる。

表１．化合物検索の推移

化合物検索	計算機	検索手法	長所○／欠点Ｘ
当初	演算速度遅い メモリー不足 貧弱なグラフィック ディスプレイ	数値／文字 データの検索	○　メモリーが少なくて済む ○　検索アルゴリズムが簡単である Ｘ　構造式のコード化は学習が必要 Ｘ　構造式からのコード化は人が行う Ｘ　総ての化合物に対応すると、 コード化のルールが複雑になる
現在	演算速度速い 大きなメモリー 強力なグラフィック ディスプレイ	結合表を用 いた検索	○　どのような化合物も対応可能 ○　研究者が親しんだ構造式の利用 ○　構造式のコード化は計算機が行う Ｘ　メモリーを多く必要とする （但し、現在の計算機では問題ない）

2．化合物の検索様式

化合物の検索様式は大きく2種類に分類される。　即ち、①完全一致検索と②部分構造検索とである。　完全一致検索は検索キーとして用いた化合物と被検索化合物とで構造式が完全に一致した化合物のみを取り出してくるものである。　部分構造検索とは化合物の部分構造を検索キーとし、被検索化合物中検索キーとして用いた部分構造をその構造式中に持つ化合物を取り出してくる検索手法てある。　当然、部分構造検索は検索部分構造と全く同じ化合物を取り出してくる時は完全一致検索となる。

利用される検索手法によるが、一般的には完全一致検索の方が部分構造検索よりも高速に検索出来る事が多い。

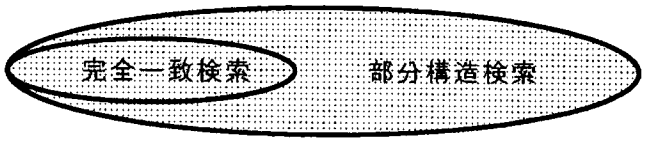


図1．完全一致検索と部分構造検索との包含関係

①　完全一致検索

完全一致検索は化合物検索における最も基本的なものである。　この完全一致検索にたいするアプローチは幾つか存在するが、これらの手法は大きく2種類に分類する事が可能である。　一つは化合物を一旦数値／文字データによる線型データに変換し、このデータを用いて検索を行うものである。　残る手法は化合物の結合表を直接用いて検索を行うもので、最近の検索はこの手法を取っている事が多い。　ここでは、これらの2手法について順を追って説明する。

(1)　変換コードを用いた検索手法

変換コードを用いた検索は、化合物を一旦数値データや文字データに変換し、その数値／文字データを検索するものである。　従って、一旦数値及び文字データに変換されればその後の検索は単なる数値／文字データの検索となり、通常用いられている検索技術の適用が可能となる。

一般的にはこのコード変換に工夫があり、線型表記の章でも述べたように一元一項対応が取られるように化合物はコード化されなければならない。　ここでは化合物の検索を最優先させた時に用いられるコード化（線型表記）を中心として述べる。　従って、説明される手法は化合物を一元一項対応で数値データに変換する事が主目的であり、一旦数値データに変換されたものを元の化合物に戻す事は考慮されていない（殆ど不可能）。

現在、このような目的で利用されている化合物のコード化手法として幾つかの手法が存在する。　ここではＭＯＲＧＡＮ名とその拡張型であるＳＥＭＡ名及び最近展開されてきたＳＭＩＬＥＳについて簡単に紹介する。　ＳＭＩＬＥＳは化合物を数値データでなく、文字データへと変換するものであり、先にのべた様々な線型表記手法に似ているが、コード化の為のルールが極めて簡単で元の構造式への変換及び結合表への変換も容易である等様々な特徴を有している手法である。

1．ＭＯＲＧＡＮ　ＮＡＭＥ（立体情報を含まない化合物の時）

1．1．ＭＯＲＧＡＮ　ＮＡＭＥ　概要

ＭＯＲＧＡＮ　ＮＡＭＥ（以下ＭＯＲＧＡＮ名と略す）は化合物のユニークナンバリング*1（化合物に対応した1個だけの番号付け）を行う為のアルゴリズムとしてＭＯＲＧＡＮにより提唱された手法である。　このＭＯＲＧＡＮアルゴリズムにより、化合物中の個々の原子について唯一種の番号付けがなされる。　この番号に基づき、原子の種類に関する情報と結合の種類に関する情報とをまとめて一つの数値データとしたものがＭＯＲＧＡＮ名である。

ＭＯＲＧＡＮ名とは化合物に一元一項対応で付けられた化合物名であるが、その名前から元の構造式を再現する事は困難である。　先に述べたようにＭＯＲＧＡＮ名の目的はユ

ニークナンバリングと検索であり、もともと構造式の再現性を目的にはしていない。

ＭＯＲＧＡＮ名　＝　ユニークナンバリング　＋　原子／結合情報

＊１）　ユニークナンバリングについて：

ユニークナンバリングとは化合物を構成する原子につけられる番号を、１化合物１通り（一元一項対応）に決定する事を意味する。
通常、化合物検索を行う時は検索キー化合物と被検索化合物との原子対応をとる事が必要となる。　人間は複数の化合物を比較する時、無意識のうちに互いに対応する原子を認識するが、計算機ではこのような事は不可能である。　従って、計算機を用いて原子に番号をつける時には、同一化合物には全く同じ番号付けがされるようにしておく事が必要である。　この一元一項対応した番号付けをユニークナンバリングという。
このユニークナンバリングに関する問題は、化合物検索を行う時に重要な問題となってくる。　つまり、完全一致検索を行う時に検索化合物と被検索化合物を構成する原子それぞれに原子番号が付けられるが、この原子番号が検索及び被検索化合物とで互いに対応がとれている事が必要となる。　この原子番号の対応が取れていない時には、例え同じ化合物であっても異なる化合物と認識されてしまう事になる。（図２参照）

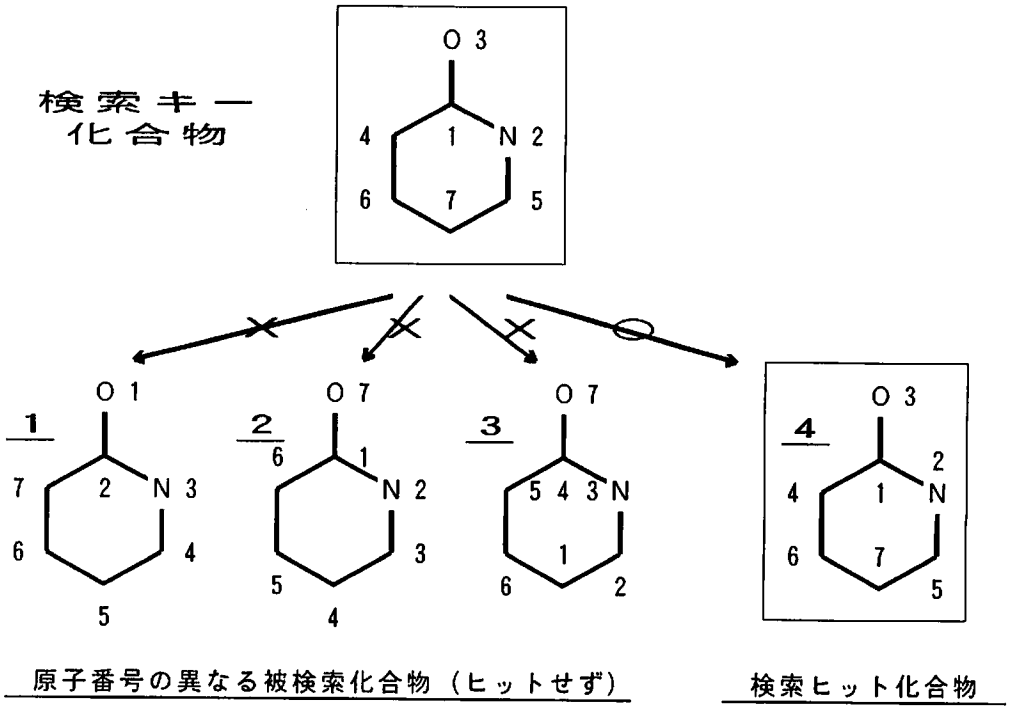


図２．　原子番号の差による化合物検索結果

化合物に付けられる原子番号が異なると、上図でも示されたように計算機の内部では全く同じ化合物であっても異なる化合物として認識される。　従って、化合物の原子に付けられる番号はユニーク（一元一項対応）でなければならない。（図中検索キー化合物と４番の化合物が同じ）　但し、ＭＯＲＧＡＮアルゴリズムによるユニーク番号付けでも限界がある。　例えば対象性の高い化合物では、このアルゴリズムで一義的に番号を付ける事は不可能であり、複数の番号付けの可能性が発生するが検索上特に大きな問題ではない。

１．２．ユニークネーミングの為のＭＯＲＧＡＮアルゴリズム

ＭＯＲＧＡＮ名を発生するには以下に示される４段階の手続きを踏まねばならない。

１．ＥＣ（Ｅｘｔｅｎｄｅｄ　Ｃｏｎｎｅｃｔｉｖｉｔｙ）の決定。
２．各ノード（ＮＯＤＥ）に対する番号付け。
３．以下のリストの作成。
　ａ．Ｆｒｏｍ　Ａｔｔａｃｈｍｅｎｔ　Ｌｉｓｔ
　ｂ．Ｒｉｎｇ　Ｃｌｏｓｕｒｅ　Ｌｉｓｔ
　ｃ．Ａｔｏｍ　Ｔｙｐｅ（Ｎｏｄｅ　Ｖａｌｕｅ）　Ｌｉｓｔ
　ｄ．Ｂｏｎｄ　Ｔｙｐｅ（Ｌｉｎｅ　Ｖａｌｕｅ）　Ｌｉｓｔ
　ｅ．Ｍｏｄｉｆｉｃａｔｉｏｎ　Ｌｉｓｔ
４．ＭＯＲＧＡＮ名の作成。

以下にはＷｉｐｋｅらの報文を基本として具体的にＭＯＲＧＡＮ名の決定についてのべる。　用いた化合物は図１に示されている。

１．３．ＭＯＲＧＡＮ名決定事例

対象化合物構造式

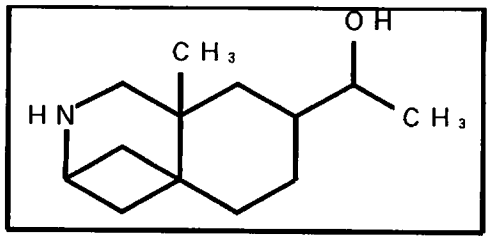


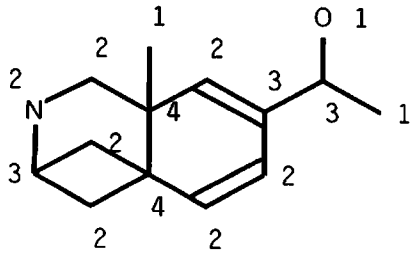
図１．ＭＯＲＧＡＮ名算出サンプル化合物

手順１：　ＥＣ（Ｅｘｔｅｎｄｅｄ　Ｃｏｎｎｅｃｔｉｖｉｔｙ）の決定

①　初期ＥＣ値の決定。

ＥＣ値は化合物中の各原子をＮＯＤＥとした時、各ＮＯＤＥ（原子）が結合しているＮＯＤＥの数（次数）のことである。

＊水素は除く



②　ＮＥＣＶ値の決定。

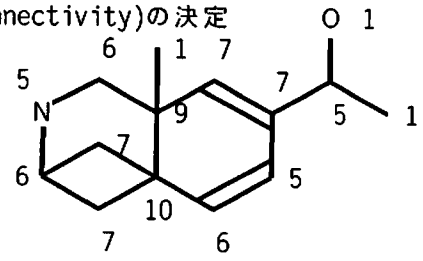
ＥＣ値の種類の数をＮＥＣＶとする。　①ではＥＣ値は１、２、３、４の４種類である。　従って、①のＮＥＣＶ値は４となる。

ＮＥＣＶ＝４

③　試行拡張分岐数（ＴＥＣ：Trial Extended Connectivity)の決定

各ＮＯＤＥに隣接しているＮＯＤＥのＥＣ値を加算し、各ＮＯＤＥに対して新たにＴＥＣを算出する。

＊末端原子は常に１とする。



④　ＮＴＥＣＶ値の決定。

ＴＥＣ値の種類の数をＮＥＣＶとする。　③ではＴＥＣ値は１、５、６、７、９、１０の６種類である。　従って、①のＮＴＥＣＶ値は６となる。

ＮＴＥＣＶ＝６

⑤　ＮＥＣＶとＮＴＥＣＶとの比較。

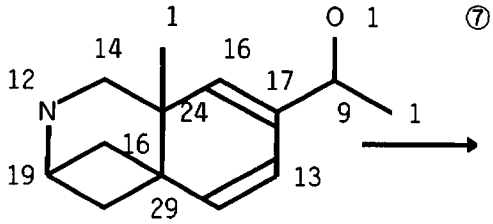
ＮＥＣＶとＮＴＥＣＶの値を比較し、

ａ．　ＮＥＣＶ≧ＮＴＥＣＶ　の時は最終ステージに行く。

ｂ．　ＮＥＣＶ＜ＮＴＥＣＶ　の時は各ＮＯＤＥのＮＴＥＣＶ値を新たにＮＥＣＶ値とみなし、ステップ③に戻る。

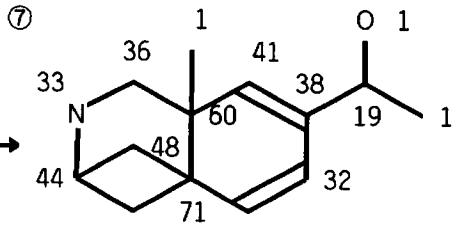
これ以降はａの状態になるまで③～⑤のステップを繰り返す。

⑥



ＮＴＥＣＶ＝１１　16　15

⑦



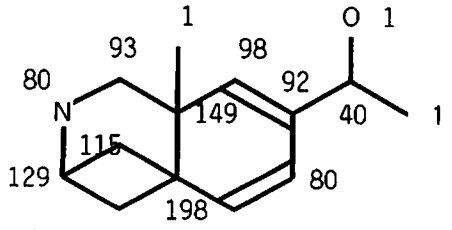
ＮＴＥＣＶ＝１２　48　42

⑧　ＥＣ値の決定。

ＮＥＣＶ（１２）≧ＮＴＥＣＶ（１１）

となるので作業を終了する。

この一つ前の時点でのＥＣ値を各ＮＯＤＥの拡張分岐数（ＥＣ）として採用する。



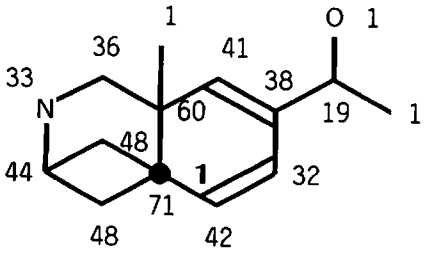
ＮＴＥＣＶ＝１１　115　103

手順２：　ＮＯＤＥの番号付け

①　１番目のノードの決定。

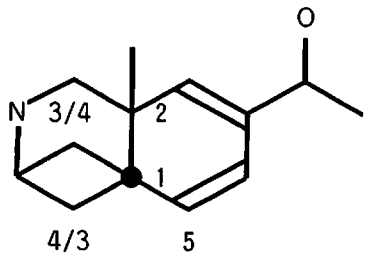
手順１の⑦で得られたＥＣ値の内、最大値を有するＮＯＤＥを選択し、１番とする。

例ではＥＣ値７１が最大なので、このＥＣ値を持つＮＯＤＥを１番とする。



② １番のＮＯＤＥのまわりのＮＯＤＥの番号決定

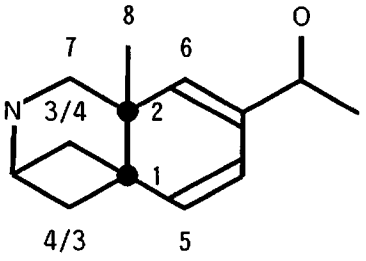
先に決定された１番のＮＯＤＥに隣接するＮＯＤＥ総てについて、ＥＣ値の大きい順に２番目以降の番号を付ける。



＊ＥＣ値が同じ時、総ての可能性について番号付けが行われる。　この時点では２種類の番号付けが可能であり、最終的には４種類の番号付けが可能となる（ＮＯＤＥ番号１に結合するＮＯＤＥ中、ＥＣ値が４８のものが２個あり、番号３と４が順位的には同値である。　また、同じＮＯＤＥに結合している測鎖のＯとＣはＥＣ値が共に１であるので番号付けの順位は同じである）。　化合物の対象性が大きくなると番号付けの可能性がおおきくなり、多数の番号付けが出てくることになる。　これらの番号付けのなかで、ＭＯＲＧＡＮ番号が最も小さくなるものが最終的なＭＯＲＧＡＮ番号として採用される。

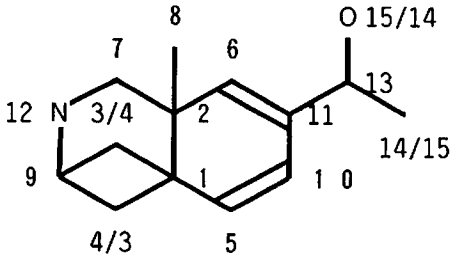
③ １番から２番のＮＯＤＥまわりへの拡張。

１番のＮＯＤＥに隣接する総てのＮＯＤＥに番号がふられたならば、次に２番のＮＯＤＥに隣接する残りのＮＯＤＥに対し、ＥＣ値の順に番号を付ける。



④ 総てのＮＯＤＥに対する番号付け。

③の手続きを化合物上の総てのＮＯＤＥに対し適用する。　総てのＮＯＤＥに対し番号を割付、番号付けを完了する。



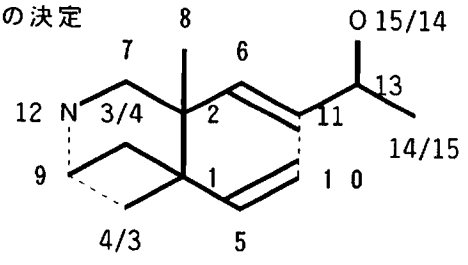
この手順２を完了した時点で与えられた化合物を構成するすべての原子について一元一項対応の番号（ユニークナンバリング）がつけられた事になる。

図からもわかるように対象性の強い化合物では番号付けの可能性が多数発生し、一元一項対応を守る事は不可能となる。　これがこのアルゴリズムの限界である。

手順３：　種々リストの作成及びＭＯＲＧＡＮ名候補の作成

① Ｆｒｏｍ　Ａｔｔａｃｈｍｅｎｔ　Ｌｉｓｔの決定

構造式中の各ＮＯＤＥの結合関係を示すリストである。　各ＮＯＤＥに番号が付けられた時の基準となったＮＯＤＥの番号が各ＮＯＤＥについて与えられる。　結果としてＦｒｏｍ　Ｌｉｓｔは化合物中の結合を定義するものとなる。



Atom Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
From List	-	1	1	1	1	2	2	2	3	5	6	7	11	13	13

＊From List は右図の実戦で描かれた結合を定義している。 点線で描かれた結合に関してはＦｒｏｍ Ｌｉｓｔでは定義されない。

② Ｒｉｎｇ Ｃｌｏｓｕｒｅ Ｌｉｓｔの決定

①の構造式で、Ｆｒｏｍ Ａｔｔａｃｈｍｅｎｔ Ｌｉｓｔでは定義されなかった点線で示される結合をこの Ｒｉｎｇ Ｃｌｏｓｕｒｅ Ｌｉｓｔで定義する。
この時は（４、９）、（９、１２）、（１０、１１）で示される。 この数値の並びをそれぞれのペアの番号を昇順に並べたものをＲｉｎｇ Ｃｌｏｓｕｒｅ Ｌｉｓｔとして採用する。 従ってこの場合は４、９、９、１２、１０、１１となる。

③ Ａｔｏｍ Ｔｙｐｅ Ｌｉｓｔの決定

原子の種類を定義するリストである。 このリストは原子に付けられた番号順に並べられる。

Atom Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Atom Type List	C	C	C	C	C	C	C	C	C	C	C	N	C	C	O

④ Ｂｏｎｄ Ｔｙｐｅ Ｌｉｓｔの決定

結合に関する情報を定義するリストである。 単結合、２重結合、３重結合、芳香族結合にそれぞれ１、２、３、４の番号を割り当てる。 このリストはＦｒｏｍ Ｌｉｓｔ とＲｉｎｇ Ｃｌｏｓｕｒｅ Ｌｉｓｔにより示される結合について定義される。

From List	-	1	1	1	1	2	2	2	3	5	6	7	11	13	13	
Ring Closure List																4,9; 9,12; 10,11
Bond Type List	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1 1

⑤ Ｍｏｄｉｆｉｃａｔｉｏｎ Ｌｉｓｔの決定

このリストは電荷、同位体、異常原子価等の記述に使用するものである。
以上のリストをまとめて一つのリストにする。

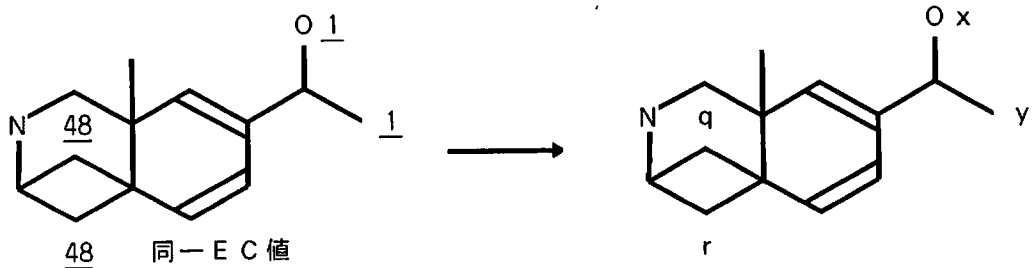
Atom Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
From List	-	1	1	1	1	2	2	2	3	5	6	7	11	13	13	
Ring Closure List																4,9; 9,12; 10,11
Atom Type List	C	C	C	C	C	C	C	C	C	C	C	N	C	C	O	
Bond Type List	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1 1

ここで C、N、Oとにそれぞれ１、２、３の番号をアサインすると、前記Ｌｉｓｔを総て結合する事によりＭＯＲＧＡＮ名が生成される。 従って、今回用いた対象化合物に対するＭＯＲＧＡＮ名の一つは以下のような数値データとなる。 但し、この時点では番号付けは絶対的なものでなく、複数のＭＯＲＧＡＮ名が発生している。 以下にこれら複数（４種類）発生されるＭＯＲＧＡＮ名の一例を示す。 このＭＯＲＧＡＮ名では酸素原子が１５番に、末端メチルが１４番にアサインされている。

候補ＭＯＲＧＡＮ名：

0101010102020203050607111313 FROM LIST	040909121011 RING CLOSURE
0101010101010101010102010103 ATOM TYPE LIST	0101010101010102020101010101010101 BOND TYPE LIST

手順４： Ｂｅｔｔｅｒ Ｎａｍｅの取り出し及びＭＯＲＧＡＮ名の最終決定



48　同一ＥＣ値　　r

手順３の時点でＭＯＲＧＡＮ名の発生が可能となる。　この時点で唯一個だけのＭＯＲＧＡＮ名が発生されていれば、その名をそのまま対象化合物のＭＯＲＧＡＮ名と採用する事が可能である。　しかし、一般的には例で示されるように複数のＭＯＲＧＡＮ名が発生されている。　つまり対象化合物には２ヵ所の同一ＥＣ値を持つ所があり、４ヵ所のＮＯＤＥが関与している（上図中番号が３、４及び１４、１５）。　従って番号付けの可能性はそれぞれの場所で２通りの可能性を持ち、総計で４種類の番号付けの可能性を含んでいる。

従って、一元一項対応の目的から、これら複数発生したＭＯＲＧＡＮ名を可能な限り少なくする事が必要である。　即ち、複数ＭＯＲＧＡＮ名が発生した時は、その中でＢＥＴＴＥＲ　ＮＡＭＥと呼ばれるものを選択する。

＊　ＢＥＴＴＥＲ　ＮＡＭＥとは、ＭＯＲＧＡＮ名を数値とした時、より小さな値を持つ名前を指している。

①　番号付け可能ケースに関する表の作成

対象化合物中の同位ＮＯＤＥ総てにアルファベットにより ＩＤ符号をアサインする。この場合はｑ、ｒとｘ、ｙとラベルする。

先ず、各ＮＯＤＥのｑ、ｒ、ｘ、ｙとが取りうる総ての番号についてまとめ、簡単な表を作成する。　この表を作成する時、ＮＯＤＥ番号は昇順に並べておく事が必要である。

番号付けのケース	NODE番号			
	3	4	14	15
ケース１	q	r	y	x
ケース２	q	r	x	y
ケース３	r	q	y	x
ケース４	r	q	x	y

②　原子の種類に関する ＩＤコードを用いた作表

続いてこの表の各ＮＯＤＥについて先にアサインされている数字（Ｃ＝１、Ｎ＝２、Ｏ＝３）を割当てて新たな表を作成する。

番号付けのケース	NODE番号			
	3	4	14	15
ケース１	1	1	1	3
ケース２	1	1	3	1
ケース３	1	1	1	3
ケース４	1	1	3	1

③　Ｂｅｔｔｅｒ　Ｎａｍｅの選択及びＭＯＲＧＡＮ名の作成

４通りのケースについて３、４、１４、１５とをまとめ、４桁の数字と考えた時、その値が最も小さいものをＢＥＴＴＥＲ　ＮＡＭＥとして採用する。

例に用いた化合物はケース１と３とが１１１３でケース２と４の１１３１よりも小さいので採用される。　ケース１と３とでは値が同じなのでこれ以上の選択は不可能であり、最終的な番号付けは２種類採用される事になる。　つまり、ｙが１４でｘが１５となり、ｑとｒは３及び４の両方採用される。　従って、最終ＭＯＲＧＡＮ名は２種類となる。

この効果はＡｔｏｍ　Ｔｙｐｅ　Ｌｉｓｔの番号で該当する炭素と酸素の順番を入れ換える事で小さな値の方を採用する事を意味している。

Atom Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Atom Type List	C	C	C	C	C	C	C	C	C	C	C	N	C	C	O	
○Atom Type List（数値）	1	1	1	1	1	1	2	1	1	1	1	1	1	1	3	（小）
Atom Type List（数値）	1	1	1	1	1	1	2	1	1	1	1	1	1	3	1	（大）

ここで用いた化合物においては、先に示した候補ＭＯＲＧＡＮ名がそのまま最終ＭＯＲＧＡＮ名として採用される。

候補ＭＯＲＧＡＮ名：
0101010102020203050607111313040909121011010101010101010101010102010103
0101010101010101020201010101010101

２．ＳＥＭＡ（立体情報を含む化合物に適用可能な名）

２．１．　ＳＥＭＡ概要

ＳＥＭＡ(STEREOCHEMICALLY EXTENDED MORGAN ALGORITHM)名はその名の通り、ＭＯＲＧＡＮ名が立体情報を含む化合物の識別が不可能であったものを、立体を含む化合物まで識別可能となるようにＭＯＲＧＡＮアルゴリズムを拡張したものである。　従ってＳＥＭＡ名とは、立体情報を含む化合物の識別を目的とするユニークネーミングの算出法を意味する。　このＳＥＭＡ名はＭＯＲＧＡＮ名の後半部分に立体に関する情報を付加させた形式を取っている。

ＳＥＭＡ名　＝　ＭＯＲＧＡＮ名　＋　立体情報に関するコード

＊ W.Todd Wipke and Thomas M. Dyott,"Stereochemically Unique Naming Algorithm", J.A.C.S., 96, 4834(1974).

２．１．１．　ＭＯＲＧＡＮアルゴリズムの立体化学への拡張

ＭＯＲＧＡＮアルゴリズムを基本とし、立体や幾何情報を含む化合物に適用可能なように拡張する時、検索で問題となる一元一項対応を守る為に化合物中に存在する立体中心*1（Stereo Center ）に関する配座情報を総て含ませる事が必要となる。

＊１．ここで定義される立体中心とは、反転（Inversion)する事で異なる立体異性体を形成する構造的特徴を意味し、化学で一般的にいわれている立体中心とは異なるものである。　ここでいう立体中心に該当するものは、化学では不斉炭素及び、炭素－炭素２重結合（炭素以外にも拡張可能）が該当する。
＊２．文献中ではヘテロ原子を含む立体に関する情報は議論の対象としていないが、容易に適用拡張可能である。

ＭＯＲＧＡＮアルゴリズムの拡張によりＳＥＭＡ（Stereocally Extended Morgan Algorithms）名へと導く事ができる。　このＳＥＭＡ名によりＭＯＲＧＡＮ名では識別不可能であった立体異性体や幾何異性体等の識別が可能となる。　しかし、このＳＥＭＡ名にも限界があり、回転異性体等の識別は不可能である。

２．１．２．　立体異性体や幾何異性体を識別する為のルール

立体情報を識別し、記号情報へと変換する目的で２種類のパリテイを定義する。　このパリテイはＥＶＥＮ及びＯＤＤ（以下適宜（Ｅ）及び（Ｏ）と略す）で表現され、化合物中の立体中心に、このどちらかのパリテイが設定される。
以下にこのＥＶＥＮ及びＯＤＤの定義について不斉炭素と２重結合の場合を例に取って説明する。

Ａ．不斉炭素の時

①　水素を含まない不斉炭素の時

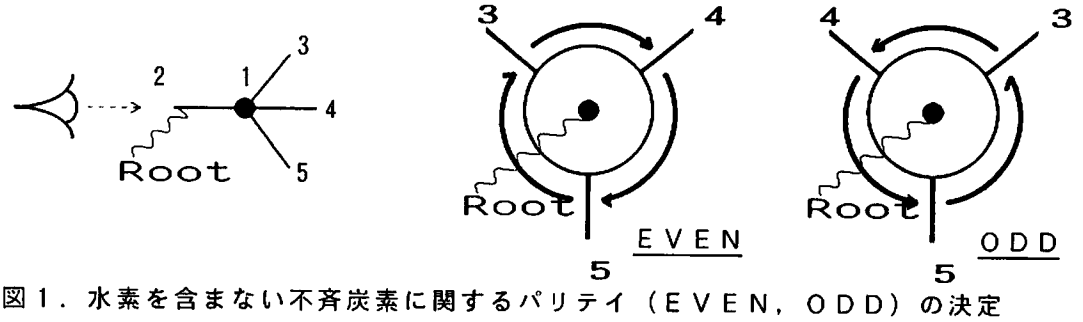


図１．水素を含まない不斉炭素に関するパリテイ（ＥＶＥＮ，ＯＤＤ）の決定

②水素を含む不斉炭素の時

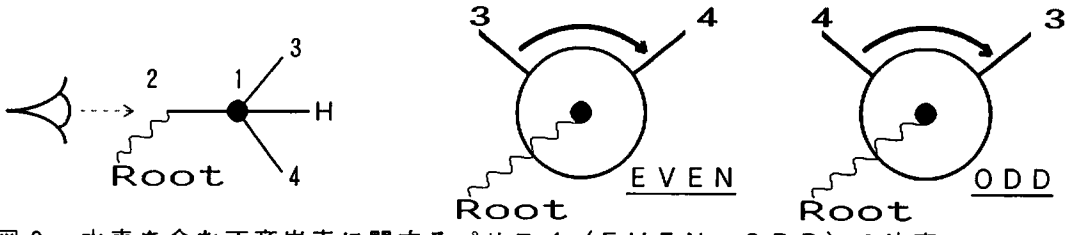


図２．水素を含む不斉炭素に関するパリテイ（ＥＶＥＮ，ＯＤＤ）の決定

図中の原子に付けられた番号はＭＯＲＧＡＮアルゴリズムにより決定された番号である。　パリテイ決定の為の手続きは、先ず目的とする立体中心に結合した原子の番号中、最も小さい番号の原子を視点とする。　この視点から立体中心原子を見た時、数字の並びが時計回りの時ＥＶＥＮ、反時計回りの時ＯＤＤと定義する。

Ｂ．２重結合の時

２重結合に関しては、その２重結合を形成する２個の原子について個別にパリテイがアサインされる。　つまり不斉炭素の時と同様、一個の２重結合原子を中心としてみた時、隣接原子の番号の並びが時計回りの時はＥＶＥＮ、反時計回りの時はＯＤＤがアサインされる。　但し、２重結合原子の隣接２原子が互いに非等価である事が前提である。
このパリテイのアサインに先立ち、２重結合を固定する事が必要となる。　この固定は２重結合を形成する２個の原子に付けられた番号に注目し、その原子番号が小さい原子を左に置き、大きい方の原子を右に置くように定義する。　このようにして固定された状態の２重結合に関しパリテイを決定する。　以下にその事例を示す。

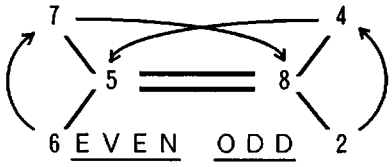


図３中、原子５に結合している隣接原子の番号は小さい方から順にみた時、時計回りである。従って、原子５のパリテイはＥＶＥＮである。また、原子８に関しては２、４、５であるので反時計回りとなりＯＤＤがアサインされる。

図３．２重結合のパリテイ定義

２．１．３．　不斉炭素及び２重結合に関するパリテイの決定事例

①　不斉炭素のパリテイ

不斉炭素及びその不斉炭素に直結する４原子にＭＯＲＧＡＮアルゴリズムにより左図の様な番号が付けられていたとすると、不斉炭素５の廻りの原子のうち最も小さな番号を持つ原子の方向に視点を置く。　この視点から７番の原子を通し、不斉炭素を眺めた時、残る３原子に付けられた番号の並びをチェックする。　この時１２～１４～１５という原子の並びは反時計回りであるので、この不斉炭素の並びはＯＤＤとなる。

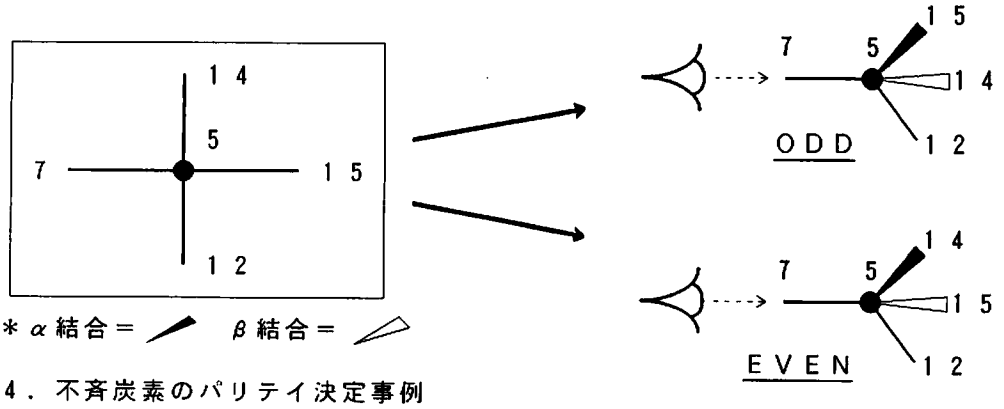


図４．不斉炭素のパリテイ決定事例

２．２．ＳＥＭＡ名によるＭＯＲＧＡＮ名限界の克服

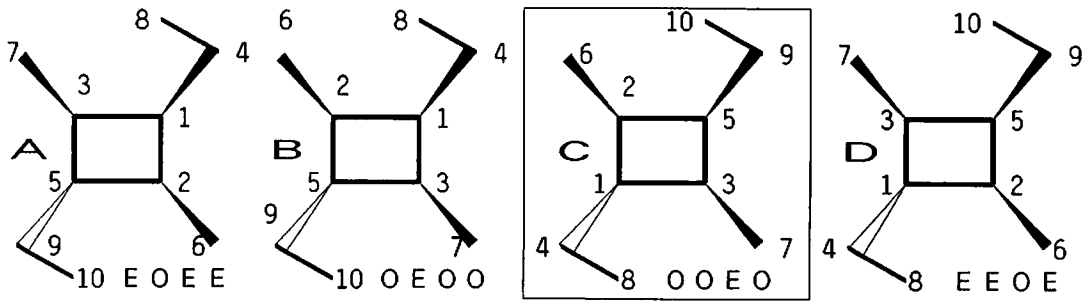


図５．ＭＯＲＧＡＮアルゴリズムにより創出された４種の原子番号

上図で示された化合物に対し、ＭＯＲＧＡＮアルゴリズムでは４種類の番号付けが可能であり、ユニークな番号をつける事は不可能である。　この問題が、先のパリテイを利用する事で解決可能となる事を示す。
先ず前記４化合物中の立体中心（１、２、３、５）についてパリテイを調べる。　決定されたパリテイに対し、ＥＶＥＮ＝２、ＯＤＤ＝１を与え、その数値の大小を比べる。

原子番号	１２３５	数値列	原子番号	１２３５	数値列
Ａ：	ＥＯＥＥ	＝　２１２２	Ｂ：	ＯＥＯＯ	＝　１２１１
Ｃ：	ＯＯＥＯ	＝　１１２１	Ｄ：	ＥＥＯＥ	＝　２２１２

これら４個の数値を比べた時、最小の数列となるＣ（１１２１）がこの化合物のユニーク名として採用する。　従って、Ｃに付けられた番号が最終番号となる。

２．３．ＭＯＲＧＡＮ名からＳＥＭＡ名の作成

２．２節でユニーク番号付けに関し、立体に関するパリテイを導入する事でＭＯＲＧＡＮアルゴリズムの限界をクリアする事が出来る事をしめした。　一旦このユニーク番号が決定されれば、この番号を基準としてＳＥＭＡ名を発生する事が可能となる。
このＳＥＭＡ名は先にも示した通り、ＭＯＲＧＡＮ名に立体に関する情報を付加する事で得られる。　この立体に関する情報は各原子毎のパリテイを数値データに変換して得られた数値列としてえられる。　２．２節の説明で用いた化合物を例に取るならば以下の様な表が得られる。　表１中、０は立体中心とは関係の無い原子を、１及び２はＯＤＤとＥＶＥＮとを示している。

表１．最終原子配置リスト

原子番号：	１	２	３	４	５	６	７	８	９	１０
パリテイ：	１	１	２	０	１	０	０	０	０	０

この数値データ列は２重結合に関する情報であれば２重結合配置リスト（Double Bond Configulation List）として、不斉炭素であれば原子配置リスト（Atom Configulation List と称される。　これらのリストをＭＯＲＧＡＮ名の最後に付加する事でＳＥＭＡ名となる。

２．４．立体化学とＳＥＭＡ名との関係

ＳＥＭＡは単に立体情報を含む化合物のユニークネーミングの発生による検索だけに利用されるわけではない。　ＳＥＭＡの特徴をいかす事で立体化学情報に関する様々な利用が可能である。　以下にはその例の幾つかについて述べる。

① ＥＮＡＮＴＩＯＭＥＲに対するＳＥＭＡ名発生

化合物が与えられた時、その化合物のパリテイを検討する事でＥＮＡＮＴＩＯＭＥＲのパリテイを決定する事が可能である。

図６．ＥＮＡＮＴＩＯＭＥＲを有する化合物

化合物Ａの立体パリテイの検討により、ＥＮＡＮＴＩＯＭＥＲである化合物Ｂの立体パリテイの決定が可能である。　先ず、化合物Ａに対しＭＯＲＧＡＮアルゴリズムによる番号付けは以下のａ～ｄで示された４種類が発生される。

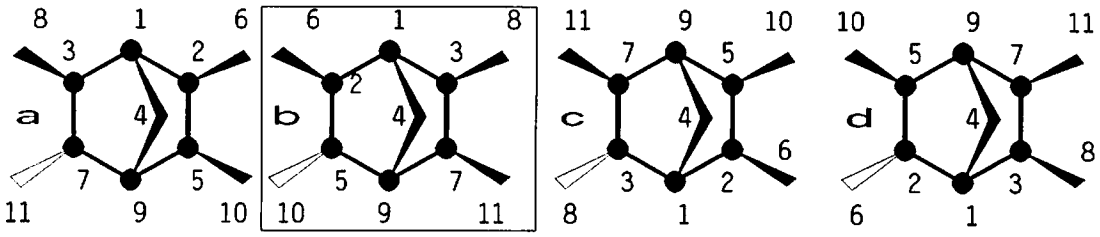


図７．ＭＯＲＧＡＮアルゴリズムにより創出された４種類の原子番号

この４種類の番号付けに従った時の立体パリテイをチェックし、その順番に並べる。

表２．図７の化合物に対する立体パリテイ

原子番号：	1	2	3	5	7	9		数値列	序列
a	E	O	E	O	O	O	=	2 1 2 1 1 1	
b	O	E	O	O	O	E	=	1 2 1 1 1 2	最低序列
c	O	E	E	E	O	E	=	1 2 2 2 1 2	
d	E	E	E	O	E	O	=	2 2 2 1 2 1	最高序列

前記４種の番号付けに対する立体パリテイは表２の様になる。　従って、このパリテイを１（ＯＤＤ）と２（ＥＶＥＮ）とで置換し、相当する数値列へと変換する。　ここで得られた数値列のうち、立体パリテイ決定に関する２．３節で示した定義より、化合物Ａに対する最終的な番号付けは値の最も小さなｂに決定される。

ＡのＥＮＡＮＴＩＯＭＥＲであるＢのパリテイは、Ａの最低及び最高序列のパリテイを比較し、両方に共通なパリテイを最低序列のものに関し反転させる事により得られる。

原子番号：	1	2	3	5	7	9		1	2	3	5	7	9
b	O	E	O	O	O	E（最低）	⟶	O	O	O	E	O	E
d	E	E	E	O	E	O（最高）	入れ換え						

この解析で得られた共通パリテイを有する原子２と原子５は次に述べるキラリテイの判定時に利用される重要な情報となるので、Reduced Set of Chiral Centers (Src) として、Ｓｒｃ＝｛２，５｝として表現する。

このＳｒｃは他の立体中心原子と区別する為、ＳＥＭＡ名の中でも特に明記して記述するようになっている。　つまり、これらＳｒｃの原子の立体パリテイの値に３を加えたもの（ＯＤＤ＝１＋３＝４，ＥＶＥＮ＝２＋３＝５）が最終原子配置リスト（Atom Configulation List）として採用される。　表３には化合物Ａ及びＢの原子配置リストが示され

ている。

表３．化合物Ａ及びＢの原子配置リスト及びＳｒｃ原子

原子番号：	1	2	3	4	5	6	7	8	9	10	11
化合物Ａ	1	5	1	0	4	0	1	0	2	0	0
化合物Ｂ	1	4	1	0	5	0	1	0	2	0	0
Src 情報		Src			Src						

② キラリテイ有無の判定

化学において化合物のキラリテイ有無を判定することが必要となる場合がある。 ＳＥＭＡ名ではＳｒｃの存在を判定する事で、化合物のキラリテイの判定を行える。

Ｓｒｃ＝０の時、化合物はアキラル（ＡＣＨＩＲＡＬ）である。
Ｓｒｃ≠０の時、化合物はキラル（ＣＨＩＲＡＬ）である。

図６の化合物はＳｒｃ＝２である為キラルである。 一方、図５の化合物は最低序列（ＯＯＥＯ）と最高序列（ＥＥＯＥ）との間で、共通なパリテイが存在しないのでＳｒｃは０であり、アキラルとなり、ＥＮＡＮＴＩＯＭＥＲは存在しない。

③ 配座異性体の識別

ＳＥＭＡ名を拡張した『拡張ＳＥＭＡ』により配座異性体の識別が可能となる。
この拡張ＳＥＭＡでは２面体角を形成する４個の原子の決定とその角度情報を決定し、ＳＥＭＡ名の後に結合する事で得られる。 この時、選択される２面体角はＳＥＭＡルールに適合して選択する（最も小さい値を有するＳＥＭＡナンバリングによる値を示す２面体）。 以下にtrans-Decalin を例に取り、手順に従って順番に説明する。

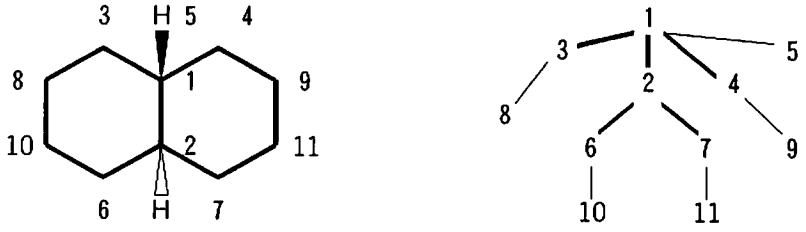


図８． trans-Decalin とスパンニングツリー

(1) スパンニングツリーの作成

trans-Decalin の配座を指定する為には原子１（ＯＤＤ）と２（ＯＤＤ）（ＳＥＭＡの番号付けにより決定された原子番号利用）に関する原子配置を特定する事で可能である。この１と２の原子に関する結合環境を明記する為にスパンニングツリーを作成する。 このスパンニングツリーは目的とする原子（１と２）を出発原子とし、３番目の原子迄の結合関係を示している。

(2) ＳＥＭＡ名に組み込む２面体角の選択

このスパンニングツリーを基本とし、原子１及び２が関係する２面体角を抽出する。個々の原子について複数の２面体角候補があるが、ここではＳＥＭＡの定義と同様に２面体角を形成する４原子の原子番号の並びが最も小さい番号を有する２面体角を採用する。これにより情報の重複を最小限にとどめる事が可能である。
例では１及び２番目の原子に関係するすべての結合（（１、２）（１、３）（１、４）（２、６）（２、７））に関し２面体角の設定が行われる。 例えば結合（１、３）に関

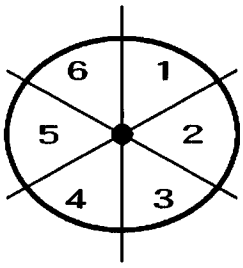
しては２面体角を形成する原子の組み合わせは（２１３８）、（４１３８）、（５１３８）の３種類あるが、この３種のうち値が最も小さいものは（２１３８）の組み合わせである。従って、結合（１、３）の設定は原子（２１３８）による２面体角の設定で充分となる。同様に残る結合について２面体角を決定する。　最終的な２面体角をＭＯＲＧＡＮ名のＦＲＯＭリスト（結合が定義される）と対照して示したものが表４である。

表４．　最終決定２面体角構成原子ＩＤリスト

原子番号	From List	２面体角構成原子	結合角情報
1			
2	1	3－1－2－6	
3	1	2－1－3－8	
4	1	2－1－4－9	
5	1		
6	2	1－2－6－10	
7	2	1－2－7－11	
8	3		
9	4		
10	6		
11	7		

以上の手続きにより多数存在する２面体角の中で、ＳＥＭＡ名に情報として与えるべき２面体角を選択する事が出来た。　続いて、これらの指定された２面体角を実際にデータとして与える事が必要となる。

(3)　２面体角情報の取り出し



２面体角の情報は角度で与える事も可能であるが、ユニークネーミングという目的から判断すれば必ずしも細かな角度情報である必要はない。　例えば、図９で示される様に角度を６等分し、この１～６の数字を角度情報として与える事で充分であり、同時に計算機による取り扱いも簡単になる。

図９．２面体角の６分割

(4)　配座リスト(Conformation List) の作成

２面体角の情報と対応する原子番号が決定されたならば、これらの情報を原子番号順に並べる。　２面体角の情報が無い原子は０を代入する。

原子番号：	1	2	3	4	5	6	7	8	9	10	11
２面体角：	0	2	3	4	0	1	1	0	0	0	0

この配座リストをＳＥＭＡ名の最後に付加する事で『拡張ＳＥＭＡ名』となり、配座異性体の識別が可能となる。

２．５．ＭＯＲＧＡＮ名、ＳＥＭＡ名及び拡張ＳＥＭＡ名との関係

化合物のユニークナンバリング及びユニークネーミング手法として３種類の手法がある事を示した。　これらは互いに独立するものでなく、ＭＯＲＧＡＮアルゴリズムを基本とし、その機能拡大により対照化合物範囲の拡大を行っている。　図１０にこれら３手法の関係を示す。

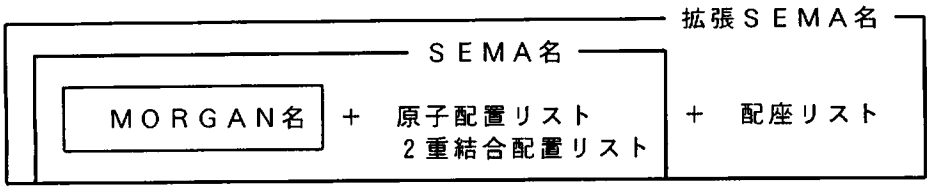


図１０．ＭＯＲＧＡＮ名、ＳＥＭＡ名及び拡張ＳＥＭＡ名との関係

２．６．ＳＥＭＡ名の限界

ＳＥＭＡ及び拡張ＳＥＭＡ名にも扱え無い化合物が存在する。　これらの手法の限界として明記する必要がある。　これらの化合物はＳＥＭＡ等の運用により解決が可能なものと、アルゴリズム上絶対的に不可能なものとにわけられる。

ＳＥＭＡ／拡張ＳＥＭＡにより表現不可能な化合物

ａ．絶対立体不斉の決定（相対立体不斉のみ識別可能）
ｂ．回転異性体の識別
ｃ．イオンの形になった化合物
ｄ．２、３量体等の複数分子で形成される化合物
ｅ．錯体化合物

これらの化合物中、ｃ、ｄ、ｅは簡単な運用によりＳＥＭＡの適用が可能である。　ａに関しては、ＳＥＭＡでは立体パリテイの決定がＭＯＲＧＡＮアルゴリズムにより付けられた原子番号を基準とし、原子量及び置換基の大小による比較ではないので絶対立体不整の決定は出来ない。　しかし、立体パリテイの決定アルゴリズムを利用し、原子量及び置換基による比較が出来るようにする事で可能となる。
ｂは現アルゴリズムでは解決不可能な問題である。