

PROLOG による概念学習の試み

——家族関係の表現演習——

市 川 新

I は じ め に

コンピュータが特定の目的のために情報処理機械として機能させるには、そのための知識を何らかのコンピュータ言語で表現することになる。いうまでもなく、情報処理機械は人間の肉体的活動の代替の機械ではなく、精神的ないし知的活動の増幅を行う機械である。「コンピュータは機械というよりも言語型思考用具である」という認識が重要である。

自然言語の特徴は曖昧性にある。人工言語の特徴は非曖昧性ないし論理的厳密性にある。コンピュータ言語は人工言語であるが、専ら数値に関わる数学言語とは異なり、非数値というより文字列を取り扱うのでその進化の方向は自然言語を意識していることになる。自然言語の場合、ある知識ないし概念をある言語系で表現することは、いかにしてその言語系がもつ曖昧性を最小限にするかに帰着する。この知的活動は論理的厳密性を追究していることになる。そうすると自然言語による思考と進化するコンピュータ言語はいずれ領域を共有する部分があることになる。そこで、本稿でも思考する人間を基準にして、コンピュータを言語型思考用具と呼ぶことにする。

この思考用具は形式的論理演算を行うにすぎないので、コンピュータ言語で表現された意味内容はその演算過程に一切の影響を与えない。言い換えれば、コンピュータ言語で表現された意味内容に曖昧性が排除されていれば、形式的論理演算を適用して得られた新しい意味解釈の内容には当然のことながら曖昧性がない。コン

ピュータ言語で知識あるいは概念を表現できれば、それを自然言語から翻訳ないし変換した人間はその意味内容について言語明瞭意味明晰の水準に達していることになる。さらに、この一連の翻訳ないし変換過程において意味の構造化ないし成分の階層化が可能であり、意味の一貫性が維持されているかぎり、記憶媒体に写像された概念の抽象化と具体化の両方向の段階的展開に限界はない。

形式的論理演算を行うにすぎない機械であるが故に、コンピュータ言語は人間の思考用具となる。この意味において求められるコンピュータ言語には、深層にあると思われる知識ないし概念をコンピュータ言語で表現することによって、表層に形として現われるまでの内省と調和した環境が具現化されていることが必要である。さらにある分野の専門家がその思考用具自体に関わる知識を必要最低限の犠牲で獲得できることが必要である。

筆者の研究室に所属する経済学部学生には、卒業論文の指導目標として、コンピュータ言語を本稿における定義に基づく思考用具として利用できることを期待している。卒論の研究対象の選択は学生に任されているので、学生の勉学環境を反映して社会問題や経済問題等多岐に亘っている。つまり、ある分野の専門家がある特定の思考用具自体に関わる知識を獲得するための基準として、学部卒業時の知識・経験を持つ者に思考の増幅ないし援助システムとして機能することを念頭においている。本稿では思考の用具としてコンピュータと対話している学生が行う概念構築の初期段階について報告する。

II PROLOG による概念モデル記述

多種多様化するコンピュータ言語には視点の異なる分類基準がある。たとえば命令型言語、関数型言語、論理型言語に分類する基準がある。人間の知的活動の一部分に限定した分野に適用する数値計算型言語あるいは事務計算型言語という分類基準もある。あるいは最近の研究成果を背景として人工知能向き言語という分類も使われはじめた。また、オブジェクト指向言語という概念が情報計算機科学者の間で注目されはじめた。このところ、この種の分類基準が一義的に定まらなくなってきたのは、広く社会で期待されるコンピュータ言語の役割の質的変化の現われと見ることができる。それは、人間の知的活動を研究対象とする認知科学や考える機械を研究対象とする情報計算機科学の研究動向に密接に関係している。すなわち、ある専門家が特定分野での専門的知識の運用能力と思考用具としてのコンピュータ言語の有用性が知られるようになったことである。それも数学者が数学言語を駆使して思考するように、概念や個別識別を文字列で表現できるコンピュータ言語あるいは因果関係の構造を直接表現できるコンピュータ言語は、非数値情報が主要部分を占める人文社会科学系の問題領域を対象とする専門家のための人工言語による思考用具として期待されはじめた。

たとえば、本来は、社会システムの動的挙動分析を目的とした連続系シミュレーション用コンピュータ言語である DYNAMO を社会学者によって理論構築のための思考用具として使う報告¹⁾がなされている。この報告には、経済・政治・文化の諸構造をコンピュータ言語で表現し、これを社会動態分析の視座構築の用具としている。具体例として K. マルクスの経済学の論理展開を写像する試みをあげている。このコンピュータ言語は因果関係を写像することによって問題の構造的な理解に有用である。しかし、その前提として定量データを必要とするので定性データの定量化という質量変換が必須である。

思考の用具とする場合には、得られた結果の質量変換、すなわち、意味解釈についてはさらに研究すべき課題が残る。この部分については別稿でとりあげたい。

コンピュータ言語の進化の過程はその有用性に早く気付いた分野の要請が形となっている歴史といえる。その結果、従来のコンピュータ言語は「計算する機械」に基づいており、DYNAMO を社会科学系で思考用具とするには数量化理論等を利用せざるを得ない。ところが、「考える機械」の研究から、述語論理学あるいは命題論理学に基づくコンピュータ言語が創造されつつある。これによって、概念や知識等抽象的意味内容を符号化すればコンピュータによって演繹推論ができるようになった。この内、述語論理のホーン節に対応したコンピュータ言語として PROLOG 処理系の多くの方言の作成²⁾が試みられている。パーソナル・コンピュータでも商用系の PROLOG システムの環境が整備されてきている。同時に多様な分野で応用研究がなされた結果、処理系が比較的単純であるにもかかわらず述語表現であることから、今までとは異なる新しい分野でこのコンピュータ言語が注目されている。述語で表現すればそれがそのままプログラムになることから、定性的情報をコンピュータ処理することができるので、非数値データを主体とした人文社会科学系の問題に直接適用できることが知られるようになった。

日本では「コンピュータ＝電子計算機」という図式が固定しているために、PROLOG システム研究プロジェクトが最先端技術プロジェクトとして推進されているにもかかわらず、本稿の分野における教育研究が多くない。PROLOG の発祥地の欧州では、社会科学の教育研究にこの新しい概念に基づく言語の可能性について研究されている。筆者は、社会システムにおけるマンマシン情報システムを研究する立場から、PROLOG をこの分野の思考用具とする試みを行っている。これは、ある領域の専門家にとってコンピュータ・システムに期待される機能ないし環境の問題となるが、専門家たる人間につ

いても当然のことであるが問題対象の中に含まれる。ある領域の専門家がコンピュータ・システムについて専門的知識や経験をもっていることを前提としないので、この問題は結局マンマシン・インターフェイスで解決すべき問題になる。筆者の研究室に所属する経済学部学生が卒業論文の作成過程においてコンピュータ言語を思考用具として利用できる環境が明らかになれば、この問題についての一つの知見が得られることになる。

そこで、学生が学部卒業論文の PROLOG プログラムとして期待される水準を事例研究するために、R. Ennals のプログラム集から経済政策の概念を PROLOG 表現³⁾したものを取りあげている。参考のため下記にそのプログラムを例示しておく。

```
deflation means raise-taxes
raise-taxes means lower-spending-power
lower-spending-power means lower-demand-for-products
lower-demand-for-products means
    less-demand-for-imported-finished-products
lower-demand-for-products means
    less-demand-for-imported-inputs-for-manufacture
lower-demand-for-products means lower-investment
less-demand-for-imported-finished-products means
    balance-of-payments-more-favourable
less-demand-for-imported-inputs-for-manufacture means
    balance-of-payment-more-favourable
lower-investment means lower-productivity
lower-productivity means less-competitiveness
less-competitiveness means balance-of-payments-less-favourable
less-competitiveness means unemployment
unemployment means lower-spending-power
x implies y if x means y
x implies y if x means z and z implies y
(x y) chain () if x means y
(x y) chain (z:X) if x means z and (z y) chain X
```

[経済政策の PROLOG 表現: 出典 文献3) pp72-74]

実際にはこのプログラムを日本語に直訳したものを学習者に提示しているが、次節で示すように PROLOG を意識した日本語表現を定式化する必要がある。それでもこの程度の内容であれば次節の演習を経験した学生は意味解釈できる。

Ⅲ PROLOG による家族関係の表現演習

PROLOG の文法は極めて単純である。次のように事実の記述と規則ないし定義の記述を行えば質問ないし推論させることができる。

[事実の記述] $P \leftarrow$

[定義の記述] $P \leftarrow Q_1 \wedge Q_2 \wedge Q_3 \cdots Q_n$

[推論の起動] $\leftarrow Q_1 \wedge Q_2 \wedge \cdots Q_n$

これらは定義記述の構文から派生したものであるから結局 1 つの構文を理解すればよいことになる。そうするとある特定の分野の専門家であれば直ちにその専門的知識の PROLOG 表現を試みることができる。しかし、卒業論文の仕上げ段階以前の学部学生にその種の専門的知識を期待することはできない。そこで家族関係を表わす概念を PROLOG で表現する演習が行われるのが通例⁴⁾である。学生が共通にもっていると期待される概念は限られ、筆者の場合も「サザエさん一家」を事例研究している。この家族構成について説明する必要がないので、学生はこの対象について知識をもっていると仮定できる。しかし、留学生はこの対象に関する知識をもっていないので新しい事例研究対象を模索しなければならないようである。

さて、PROLOG では文字列が基本データ構造であるので概念の符号化に自然言語を積極的に流用できる。形式的論理演算が適用されるだけであるので、そこに使用される語彙群には曖昧性は許されない。従って、語ないし語句のもつ意味成分の多様な関係に配慮しなければならない。コンピュータ応用を行っている言語学では、英語の意味構造は研究例が多いのでこの種の問題に参考になる。それに対して、日本語の意味構造の研究は行われている⁵⁾が限定される。しかし、親族名称については言語間の差違がほとんどないといってよいであろう。そこで、英語における成分分析例⁶⁾を手掛かりとして、代表的プログラム例を検討することにする。

1) 事実関係の記述段階

補足資料 I に示す学生 H のプログラムは概

念構築の初期段階の典型的事例である。プログラム H は全て事実に関する記述である。PROLOG の構文は前節で述べたが、その構成要素は次の標準形式で表現する。

述語 (引数₁, 引数₂, 引数₃, ……)

引数が事物を表わすとすれば述語は事物相互の関係を表現する。一般に引数の個数が 1 個の場合はその事物の特徴を表現する。そこで、いくつかの記述の意味は次のように解釈できる。ただし、このプログラムの記述者が定めた意味とは異なることがある。

○述語表現 「意味解釈」

- ① otoko (katsuo) 「カツオは男だ」
- ② onna (sazae) 「サザエは女だ」
- ③ toshi (wakame, 4) 「ワカメは 4 歳だ」
- ④ oyako (masuo, tarao) 「マスオとタラオは親と子だ」
- ⑤ tsuma (sazae, masuo) 「サザエとマスオは妻と夫だ」
- ⑥ chichi (namihei, katsuo) 「ナミヘイとカツオは父と子だ」
- ⑦ haha (funa, wakame) 「フナとワカメは母と子だ」
- ⑧ musuko (tarao, sazae) 「タラオとサザエは息子と親だ」
- ⑨ musume (wakame, namihei) 「ワカメとナミヘイは娘と親だ」
- ⑩ mago (tarao, fune) 「タラオとフネは孫と祖父か祖母だ」

これらの述語例から、プログラム H は家族名称の全てを状況関係が抽象化されないまま実在する事実に基づいていることが分かる。①の述語は事物の特徴を素直に表現したものであり、この場合は性別について記述している。⑤では家族名称として片方向を意味する述語名が名付けられている。意味解釈では両方向として理解している。しかし、プログラム H では物語登場人物についてクロス・ネットワーク的に概念化を行い、それに名前を付けている可能性がある。もし、そうすると⑤では単に家族名称の妻の事実関係を記述したにすぎないことになる。

いずれ⑤に対応して夫の事実関係を記述することになれば、プログラム H では状況を記述した段階であり、まだ概念の階層性には気付いていないと言えよう。

2) 抽象的規則の記述段階

補足資料 II に示す学生 T のプログラムは概念構築の中期段階の典型的事例である。プログラム T は事実に関する記述と規則ないし定義に関する記述に分離しはじめたことを示している。規則ないし定義は次の標準形式で表現する。

述語 (引数₁, 引数₂, 引数₃, ……): -

述語 (引数₁, 引数₂, 引数₃, ……), 述語 (引数₁, 引数₂, 引数₃, ……), ……

この表現は事物と事物の関係に関する一般的な意味をもつ文である。この文は現実の状況が非実在化したものであり抽象的概念を表現できる。そこで、いくつかの記述の意味は次のように解釈できる。ただし、このプログラムの記述者が定めた意味とは異なることがある。

○述語表現 「意味解釈」

① oya (X, Y): - chichi (X, Y)

「X と Y が親子であるには、X と Y が父と子であればよい」

② haha (X, Y): - chichi (Z, Y), otto (Z, X)

「X と Y が母と子であるには、Z と Y が父と子であり、Z が X の夫であればよい」

③ musuko (X, Y): - oya (Y, X), otoko (X)

「X と Y が息子と親であるには、Y と X が親と子であり、X が男であればよい」

④ sobo (X, Y): - oya (Z, Y), oya (X, Z), onna (X)

「X と Y が祖母と孫であるには、Z と Y が親と子であり、X と Z が親と子であり、X が女であればよい」

⑤ mago (X, Y): - sofu (Y, X)

「X と Y が孫と祖父母であるには、Y と X が祖父と孫であればよい」

⑥ ane (sazae, katsuo)

「サザエとカツオは姉妹兄弟である」

⑦ imouto (X, Y): - ane (Y, X), onna (X)

「X と Y が妹と兄姉であるには、Y と X が姉妹兄弟であり、X が女であればよい」

プログラム H とプログラム T との本質的違いは情況関係から抽象化が始まったことにある。実在性のある記述から非実在性の記述に帰納推論が働いたことになる。①②③にみられるように、親族名称としての父は男性を表わし、母は女性を表わすことが言語で表現できるまでに達していることが分かる。さらに、性別は同じであるが、父という語は息子という語と祖父という語と世代の違いということで対比できることにもほぼ気付いているようである。また、述語の意味は単方向の関係を示しているものと思われる。たとえば⑤の意味解釈では、孫と祖父母の關係に名付けているとしているが、プログラム T の記述者はただ単に「X は Y の孫である」を陽に表現したにすぎないようである。「Y は X の祖父母である」ということは気付いていない。さらに、「X と Y は兄弟姉妹である」という概念は事実として記述されている。「X と Y は兄弟姉妹である」という概念を定義として記述するためには直系関係と個人の年齢が事実として記述されているか、あるいは推論できればよい。プログラム T では直系関係の記述はされているが年齢に関する記述は見当たらない。そのため、本来、定義として記述できるものを事実として記述したものと思われる。

3) 概念体系の記述段階

補足資料Ⅲに示す学生 I のプログラムは概念構築の最終段階の典型的事例である。プログラム I は事実に関する記述と規則ないし定義に関する記述が分離しているだけでなく、概念体系の冗長性を排除しようとしていることが読み取れる。いくつかの記述の意味は次のように解釈できる。ただし、このプログラムの記述者が定めた意味とは異なることがある。

○述語表現 「意味解釈」

① otoko (namihei) 「ナミヘイは男だ」

② onna (fune) 「フネは女だ」

③ otto (namihei, fune) 「ナミヘイとフネは夫と妻だ」

④ chichi (namihei, sazae) 「ナミヘイとサザエは父と子だ」

⑤ nenrei (sazae, 24) 「サザエの年齢は24歳だ」

⑥ tsuma (X, Y) : - otto (Y, X)

「X と Y は妻と夫であるには、Y と X は夫と妻であればよい」

⑦ haha (X, Y) : - tsuma (X, Z), chichi (Z, Y)

「Y と Y は母と子であるには、X と Z は妻と夫であり、Z と Y は父と子であればよい」

⑧ kyoudai (X, Y) : - oyako (Z, X), oyako (Z, Y), ne (X, Y), nenrei (Y, A), nenrei (X, B), gt (A, B)

「X と Y は兄弟姉妹であるには、Z と X は親子であり、Z と Y は親子であり、X と Y は別人であり、X の年齢が Y の年齢より高年齢であればよい」

プログラム H およびプログラム T と比較すると、プログラム I は事実として記述すべきことと定義として記述すべきこととの分離がほぼ完全になされている。プログラム I では親族名称に関する概念が構築されている。性別、生年月日、世代、直系という基本的関係を発見すれば、その他の親族名称はそれらに基づいて抽象的に定義できることに気付いている。たとえば、⑧の兄弟姉妹の定義に関する述語は、⑤の述語による年齢関係により順序関係を表わすことが分かる。⑦の母と子の定義に関する述語は④の父と子の述語と⑥の婚姻を表わす述語の連言により定義されていることが分かる。いずれにせよ、基本的関係を発見できることをプログラム I は示している。

概念学習ないし構築にコンピュータ言語を思考用具とした場合、概ね、事実関係の記述段階、抽象的規則の記述段階、概念体系の記述段階の3段階を経過するようである。コンピュータ言語である PROLOG との対話は、自分自身との対話である。そして、性別、生年月日、世代、直系という基本的関係からより上位の概念が定

義できることを発見する。本稿では、親族名称に関して言語間の差違がほとんどないと仮定した。しかし、基本関係を記述する場合、文化圏の価値体系が記述内容に影響を与えるのは自然であろう。すなわち、婚姻の定義と親子の定義には2通りの記述ができる。1つは本稿のプログラム例のように「父と子の関係の事実表現、その父と婚姻関係の定義から導出される母と子の事実関係」の概念と、他方は「母と子の関係の事実表現、その母と婚姻関係の定義から導出される父と子の事実関係」の概念である。アメリカでは日本の戸籍と対比できる情報管理はなされていないため、日常生活で同質の属性を必要とする書類では母親名を書かせるようである。これは、母と子の関係の方が実在性の強いことを意味しているかもしれない。いずれにせよ、形式論理演算を行うにすぎないコンピュータ・システムは、言語をもつことによって思考用具になる。

Ⅳ お わ り に

5年間の実践的教育の経験によって、述語表現と事実・規則・定義に関する5時間程度の知識教育があれば学習者は自律的にコンピュータ・システムと対話できることが分かっている。一旦、このPROLOG言語の環境に入れば、学習者には、プログラミングという能動的行為によって、漠とした認識から論理的矛盾を排除し、かつ、述語論理に基づく閉じられた概念を事実たる部分と定義たる部分とに明確に区別することが求められる。このことは前節での例示によって知られるようにほぼ目標を達成できる。

学部卒業論文の結論部程度にみられる論理展開の文書資料からその前提となった概念を抽出するという当面の学習目標には残念ながら達していない。しかし、20名の学習者がいるとすればその内の2～4名は多次元関係表までは完成できるようである。教授側の問題点は、学習者が経済政策のPROLOG言語表現を事例研究することを前提としているが、このための環境が不備であると思われる。具体的には社会科学の

問題対象を記述するのに不自由しない程度の制限日本語の形式化が求められている。外交政策等の英語文書資料からの論理・因果関係にかかわる部分を抽出するコーディング規則が報告⁷⁾されている。さらに、このためのコンピュータ援助意思決定システムの研究⁸⁾がなされている。文意味要素の階層関係が構文上静的かつ明示的傾向がある英語文書においても英語構文の反映したこの種のコーディング規則が必要とされる。同様に日本語文章資料からの論理・因果関係の抽出においても、日本語の文意味要素の統語にみられる自由度を反映したコーディング規則及びそれを知識として内蔵したコーディング援助システムが求められる。

参 考 文 献

- 1) Hanneman, R. A. (1988). Computer-assisted Theory Building, Sage Publications.
- 2) 市川 新(1986). 「移植性を意識した basic-PROLOG システム」, 流通経済大学論集, 20(1-4): 37-48.
- 3) Ennals, J. R. (1985). Artificial Intelligence: Applications to Logical Reasoning and Historical Research, Ellis Horwood.
- 4) 草薙 裕(1985). 自然言語とコンピュータ言語, 講談社.
- 5) 望主雅子(1986). 「引照関係に基づく辞書語釈の考察」, 東京女子大学日本文学, (66): 87-99.
- 6) 升川 潔他訳(1977). 意味の構造—成分分析, 研究社. [Nida, E. A. (1975). Componential Analysis of Meaning, Mouton & Co. N. V., Publishers.]
- 7) Wrightson, M. T. (1976). 'The Documentary Coding Method,' in Structure of Decision: The Cognitive Maps of Political Elites, Princeton Univ.
- 8) Nakamura, K. et al. (1982). 'Decision Support Using Causation Knowledge Base,' IEEE Trans. Sys., Man, Cybern. SMC-12(6): 765-777.

補足資料

I 学生 H の全プログラム・リスト

otoko(katsuo).
onna(wakame).
otoko(tarao).
otoko(masuo).
otoko(namihei).
onna(fune).
onna(sazae).
toshi(namihei,50).
toshi(fune,50).
toshi(masuo,30).
toshi(sazae,28).
toshi(katsuo,13).
toshi(wakame,10).
toshi(tarao,4).
oyako(namihei,sazae).
oyako(namihei,katsuo).
oyako(namihei,wakame).
oyako(fune,sazae).
oyako(fune,katsuo).
oyako(fune,wakame).
oyako(masuo,tarao).
oyako(sazae,tarao).
otto(namihei,fune).
otto(masuo,sazae).
tsuma(fune,namihei).
tsuma(sazae,masuo).
chichi(namihei,sazae).
chichi(namihei,katsuo).
chichi(namihei,wakame).
chichi(masuo,tarao).
haha(fune,sazae).
haha(fune,katsuo).
haha(fune,wakame).
haha(sazae,tarao).
musuko(katsuo,namihei).
musuko(katsuo,fune).
musuko(tarao,masuo).
musuko(tarao,sazae).
musume(sazae,namihei).
musume(wakame,namihei).
musume(sazae,fune).
musume(wakame,fune).
mago(tarao,namihei).
mago(tarao,fune).
ani(katsuo,wakame).
ane(sazae,katsuo).
ane(sazae,wakame).
otouto(katsuo,sazae).
imouto(wakame,sazae).

imouto(wakame,katsuo).
sofu(namihei,tarao).
oba(fune,tarao).
oji(katsuo,tarao).
oba(wakame,tarao).
kyoudai(sazae,katsuo,wakame).
oi(katsuo,wakame).
oi(tarao,katsuo).
oi(tarao,wakame).
mei(wakame,masuo).

II 学生 T の全プログラム・リスト

otoko(ナミヘイ).
otoko(マスオ).
otoko(カツオ).
otoko(タラオ).
onna(フネ).
onna(サザエ).
onna(ワカメ).
oya(X,Y) :-
 chichi(X,Y).
oya(X,Y) :-
 haha(X,Y).
chichi(ナミヘイ,サザエ).
chichi(ナミヘイ,カツオ).
chichi(ナミヘイ,ワカメ).
chichi(マスオ,タラオ).
haha(X,Y) :-
 chichi(Z,Y),
 otto(Z,X).
otto(ナミヘイ,フネ).
otto(マスオ,サザエ).
tsuma(X,Y) :-
 otto(Y,X).
musuko(X,Y) :-
 oya(Y,X),
 otoko(X).
musume(X,Y) :-
 oya(Y,X),
 onna(X).
sofu(X,Y) :-
 oya(Z,Y),
 oya(X,Z),
 otoko(X).
sobo(X,Y) :-
 oya(Z,Y),
 oya(X,Z),
 onna(X).
mago(X,Y) :-
 sofu(Y,X).
mago(X,Y) :-
 sobo(Y,X).

```

ane(サザ エ, カツオ).
ane(サザ エ, ワカメ).
ani(カツオ, ワカメ).
imouto(X,Y) :-
    ane(Y,X),
    onna(X).
imouto(X,Y) :-
    ani(Y,X),
    onna(X).
otouto(X,Y) :-
    ane(Y,X),
    otoko(X).
oji(X,Y) :-
    oya(Z,Y),
    ane(Z,X),
    otoko(X).
oba(X,Y) :-
    oya(Z,Y),
    ane(Z,X),
    onna(X).
oi(X,Y) :-
    oji(Y,X).
oi(X,Y) :-
    oba(Y,X).
girichichi(X,Y) :-
    tsuma(Z,Y),
    oya(X,Z),
    otoko(X).
girihaaha(X,Y) :-
    tsuma(Z,Y),
    oya(X,Z),
    onna(X).
giriiani(X,Y) :-
    ane(Z,Y),
    otto(X,Z),
    otoko(X).
giriouto(X,Y) :-
    giriiani(Y,X),
    otoko(X).
giriimouto(X,Y) :-
    giriiani(Y,X),
    onna(X).
girimusuko(X,Y) :-
    tsuma(Z,X),
    oya(Y,Z),
    otoko(X).

onna(フネ).
onna(サザ エ).
onna(ワカメ).
otto(ナミヘイ, フネ).
otto(マスオ, サザ エ).
chichi(ナミヘイ, サザ エ).
chichi(ナミヘイ, カツオ).
chichi(ナミヘイ, ワカメ).
chichi(マスオ, タラチャン).
girichichi(X,Y) :-
    tsuma(Z,Y),
    chichi(X,Z).
nenrei(ナミヘイ, 50).
nenrei(フネ, 48).
nenrei(マスオ, 25).
nenrei(サザ エ, 24).
nenrei(カツオ, 11).
nenrei(ワカメ, 9).
nenrei(タラチャン, 4).
tsuma(X,Y) :-
    otto(Y,X).
haha(X,Y) :-
    tsuma(X,Z),
    chichi(Z,Y).
girihaaha(X,Y) :-
    tsuma(X,Z),
    girichichi(Z,Y).
oyako(X,Y) :-
    chichi(X,Y).
oyako(X,Y) :-
    haha(X,Y).
giriyoako(X,Y) :-
    tsuma(Z,Y),
    oyako(X,Z).
kyoudai(X,Y) :-
    oyako(Z,X),
    oyako(Z,Y),
    ne(X,Y),
    nenrei(X,A),
    nenrei(Y,B),
    gt(A,B).
giriokyoudai(X,Y) :-
    tsuma(Z,X),
    kyoudai(Z,Y).
ozi(X,Y) :-
    otoko(X),
    oyako(Z,Y),
    kyoudai(Z,X).
oba(X,Y) :-
    onna(X),
    oyako(Z,Y),
    kyoudai(Z,X).
oi(X,Y) :-
    otoko(X),

```

Ⅲ 学生Ⅰの全プログラム・リスト

```

otoko(ナミヘイ).
otoko(マスオ).
otoko(カツオ).
otoko(タラチャン).

```

```
    ozi(Y,X).
oi(X,Y) :-
    otoko(X),
    oba(Y,X).
musuko(X,Y) :-
    otoko(X),
    oyako(Y,X).
musume(X,Y) :-
    onna(X),
    oyako(Y,X).
girimusuko(X,Y) :-
    otoko(X),
    giriroyako(Y,X).
sohu(X,Y) :-
    otoko(X),
    oyako(X,Z),
    oyako(Z,Y).
sobo(X,Y) :-
    onna(X),
    oyako(X,Z),
    oyako(Z,Y).
mago(X,Y) :-
    sohu(Y,X).
```

```
mago(X,Y) :-
    sobo(Y,X).
ane(X,Y) :-
    onna(X),
    kyoudai(X,Y).
ani(X,Y) :-
    otoko(X),
    kyoudai(X,Y).
giriani(X,Y) :-
    otoko(X),
    girikyoudai(X,Y).
otouto(X,Y) :-
    otoko(X),
    kyoudai(Y,X).
imouto(X,Y) :-
    onna(X),
    kyoudai(Y,X).
giriouto(X,Y) :-
    otoko(X),
    girikyoudai(Y,X).
giriimouto(X,Y) :-
    onna(X),
    girikyoudai(Y,X).
```