

論理プログラミングによる株式会社の機関設計

犬童健良

2006. 5. 18 (2006. 7. 3 修正)

概要

本論文では、論理プログラミング言語 PROLOG を用いて、会社法によって認められた株式会社の機関設計のルールを表現した。このコードを実行し、すべての可能な機関設計パターン 39 通りを生成できた。また、日本語による表示を含めて、指定の条件に合った機関設計を選択する方法、さらに条文に対する翻案の論理的整合性を検証する方法について論じた。

1. はじめに

株式会社の機関設計は、今年 5 月に施行された会社法[1]によって、定められている。会社法においては、中小会社の機関設計の自由度が増し、また最適な機関設計を選択するという、じゅうらいになかった問題が生じた。会社法の文献や関連ウェブサイトで、これらの機関設計の可能なパターンは 39 通りあり、図 1 に示すような原則から導かれると説明されている[2]。本論文では、論理プログラミング言語 PROLOG[3]を用いて、機関設計にかんする、以下の基礎的な問題を解いてみたい。

- 機関設計のルールを PROLOG のコードによって実装すること。
- 8 つの原則をすべて満たす機関設計 39 通りを、論理的に正しく導出すること。
- 機関設計を選択する問題（機関設計の最適化問題）を解くこと。
- 法令文章やその翻案の論理的整合性を検証すること。

ところで PROLOG は、1980 年代の AI（人工知能）研究、とりわけ自然言語処理の分野で注目された。¹ その法令文書や法的推論への応用は、やはり早くから注目されていたが、現代では

¹ 日本監査役協会によると、法制審議会第 56 号諮問答申「会社法制の現代化に関する要綱」に基づく。

企業の知財管理や企業統治への応用も考えられる。本論文はこのジャンルへのささやかな応用を試みる。後出の PROLOG コードは、再帰を用いず、バックトラックも複雑ではない平易なものであるが、現実的な有用性があると考えられる。

PROLOG プログラムは事実、ルール、ゴールの3種類の節形式(caluses)からなる。PROLOG システムは、指定されたゴールが、あらかじめ内部データベースに登録したルールと事実の集合を制約条件として満たす、論理的に可能な解（充足可能なモデル）が見つかるまで、項のパターンマッチング（単一化）と系統的な試行錯誤（バックトラック）を繰り返し用いて追求する。

- ① すべての株式会社には、株主総会のほか、取締役を設置しなければならない（326条1項）。
- ② 取締役会を設置する場合には、監査役（監査役会を含む）又は三委員会等のいずれかを設置しなければならない。ただし、大会社以外の譲渡制限会社（すべての種類の株式が譲渡制限株式である株式会社）において、会計参与を設置する場合には、この限りでない（327条）。
- ③ 株式譲渡制限会社以外の株式会社には、取締役会を設置しなければならない（327条1項）。
- ④ 監査役（監査役会を含む）と三委員会等とをともに設置することはできない（327条4項）。
- ⑤ 取締役会を設置しない場合には、監査役会及び三委員会等を設置することができない（327条1項2号・3号）。
- ⑥ 会計監査人を設置するには、監査役（監査役会を含む）又は三委員会等（大会社であって株式譲渡制限会社でない株式会社にあつては、監査役会又は三委員会等）のいずれかを設置しなければならない（327条3項・5項）。
- ⑦ 会計監査人を設置しない場合には、三委員会等を設置することはできない（327条5項）。
- ⑧ 大会社には、会計監査人を設置しなければならない（328条）。

図 1. 機関設計の原則

（出典：日本監査役協会監査法規委員会(2005),「監査役からみた会社法の捉え方」, p.21, 資料1.）

PROLOG 言語の特質として、たんにプログラミング言語として、コード化ないし実装に用いられるだけではなく、それ自体が、論理的な意味を表すものである（主張1）。また実装されたコードは、PROLOG システムの定理自動証明メカニズム（導出原理）により、実行可能である。それによって論理的モデルの正しさを実験的に検証（証明）することができるわけであるから、数学的言語による定式化と証明の組合せに代替しうる（主張3）。

本論文の以降の部分では、上で述べた3つの主張を背景に、PROLOG を用いて、冒頭で述べた4つの問題を、順に解いていく。

冒頭の4つの課題に立ち戻ると、最初の2つは、それなりに手間はかかるが易しい練習問題である（第2節と第3節）。3番目は、一種の最適化問題であり、それゆえユーザ、例えば創業者あるいは親会社の、プリファレンス（目的や制約条件）をどのようにモデリングするかが問題になる。本論文では、リストを用いた簡単な抽出方法を論じる（第4節）。最後の部分は、機関名などを日本語で扱う方法に始まり、法令文とその翻案の正当性や冗長性を検証する問題に言及する（第5節）。また究極的に、前の3要素と合わせて、これらはより広い意味で制度設計を支援するシステム（IDSS ; Institutional Design Supporting Systems）の基礎をなすものと考えられる。最後にまとめとする（第6節）。

表 1. 機関設計原則に対応する PROLOG プログラム

節の種類	述語名/引数の個数	意味
事実	kabusiki_kaisha/2	株式会社の種類
事実	kaisha_no_kibo/2	会社の規模
事実	setti/2	各機関を設置する(yes)か、しない(no)かの二者択一
ルール	kabusiki_gaisha_no_kikan_sekkei/2	機関設計の全体にかんするルール
ルール	torisimari-yaku/1	取締役の設置にかんするルール
ルール	torisimari-yaku-kai/2	取締役会の設置にかんするルール
ルール	check-kikan/2	チェック機関の設置にかんするルール
ルール	kansa-yaku-kai/2	監査役会の設置にかんするルール
ルール	kaikai-kansa-nin/2	会計監査人の設置にかんするルール
ルール	kabunusi-sokai/1	株主総会の設置にかんするルール
ルール	sekkei_rule/1	各種機関にかんする個別ルール・条文との対応

```

kabusiki_kaisha(kokai, '公開会社').
kabusiki_kaisha(joto-seigen, '株式譲渡制限会社').

kaisha_no_kibo( dai, '大会社').
kaisha_no_kibo( chu-sho, '中小会社').

setti(_, yes).
setti(_, no).

kabusiki_gaisha_no_kikan_sekkei((S,T), [A,B,C,D,E,E1,F,G,H,I]) :-
    kabusiki_kaisha(A,S),
    kaisha_no_kibo( B,T),
    torisimari-yaku(C),
    torisimari-yaku-kai(A,D),
    check-kikan([A,B,D], [E,F,G]),
    kansa-yaku-kai([D,E,F],E1),
    kaikei-kansa-nin([A,B,E,E1,F],H),
    kabunusi-sokai(I).

torisimari-yaku(C) :-
    setti(torisimari-yaku,C),
    sekkei_rule(326-1, torisimari-yaku,C).

kabunusi-sokai(I) :-
    setti(kabunusi-sokai, I),
    sekkei_rule(326-1, kabunusi-sokai,I).

torisimari-yaku-kai(A,D) :-
    setti(torisimari-yaku-kai, D),
    sekkei_rule(327-1,_,(A,D)).

check-kikan([A,B,D], [E,F,G]) :-
    setti(kansa-yaku, E),
    setti(san-iinkai, F),
    setti(kaikei-sanyo, G),
    sekkei_rule(327,_,([D,A,B], [E,F,G])),
    sekkei_rule(327-4, \+ (kansa-yaku,san-iinkai), [E,F]),
    sekkei_rule(327-1-2-3, san-iinkai->_,(F,D)).

kansa-yaku-kai([D,E,F],E1) :-
    setti(kansa-yaku-kai, E1),
    (E,E1) \= (no, yes),
    sekkei_rule(327-4, \+ (kansa-yaku-kai,san-iinkai), [E1,F]),
    sekkei_rule(327-1-2-3, kansa-yaku-kai->_,(E1,D)).

kaikei-kansa-nin([A,B,E,E1,F],H) :-
    setti(kaikei-kansa-nin, H),
    sekkei_rule(327-3,_,(H, [E,F])),
    sekkei_rule(327-3-5,_,([H,A,B], [E1,F])),
    sekkei_rule(327-5,_,(H,F)),
    sekkei_rule(328-1,_,(B,H)).

```

図 2. PROLOG によって機関設計原則を表現する：機関ごとの設計ルール

```

sekkei_rule(326-1,torisimari-yaku,yes).
sekkei_rule(326-1,kabunusi-sokai,yes).

sekkei_rule(327, check-kikan, ([no|_], _)).
sekkei_rule(327, check-kikan, ([yes,joto-seigen,B], [no,no,yes])):-
    B \= dai.
sekkei_rule(327, check-kikan, ([yes|_], [_yes,_])).
sekkei_rule(327, check-kikan, ([yes|_], [yes,_,_])).

sekkei_rule(327-1,torisimari-yaku-kai,(A,yes)):-
    A \= joto-seigen.
sekkei_rule(327-1,torisimari-yaku-kai,(joto-seigen,yes)).
sekkei_rule(327-1,torisimari-yaku-kai,(joto-seigen,no)).

sekkei_rule(327-1-2-3, kansa-yaku-kai->torisimari-yaku-kai,(no,no)).
sekkei_rule(327-1-2-3, san-iinkai->torisimari-yaku-kai,(no,no)).
sekkei_rule(327-1-2-3, _->torisimari-yaku-kai,(_,yes)).

sekkei_rule(327-2, check-kikan, (no, _)).
sekkei_rule(327-2, check-kikan, (yes, EFG)):-
    EFG \= [no,no,no].

sekkei_rule(327-3, kaikei-kansa-nin->kansa-yaku;san-iinkai,(yes,EF)):-
    EF \= [no,no].
sekkei_rule(327-3, _,(no,_)).

sekkei_rule(327-3-5, kaikei-kansa-nin->P,([yes,A,dai],EF)):-
    P = kansa-yaku-kai;san-iinkai,
    A \= joto-seigen,
    EF \= [no,no].
sekkei_rule(327-3-5, _->kansa-yaku-kai;_,([yes,joto-seigen,_,_])):-!.
sekkei_rule(327-3-5, _->kansa-yaku-kai;_,([yes,_,B],_)):- B \=dai,!.
sekkei_rule(327-3-5, _->kansa-yaku-kai;_,([no|_],_)).

sekkei_rule(327-4, \+ (X,san-iinkai),EF):-
    member( X,[kansa-yaku,kansa-yaku-kai]),
    EF \= [yes,yes].

sekkei_rule(327-5, \+ kaikei-kansa-nin-> \+ san-iinkai,(no,no)).
sekkei_rule(327-5, _,(yes,_)).

sekkei_rule(328-1, dai-gaisha->kaikei-kansa-nin,(dai,yes)).
sekkei_rule(328-1, _,(chu-sho,_)).

```

図 3. PROLOG によって機関設計原則を表現する：個別のルール

```

:- dynamic sekkei_data/3,sekkei_data_id/1.

init_sekkei_data:-
    abolish( sekkei_data/3),
    abolish( sekkei_data_id/1),
    assert(sekkei_data_id(0)).

update_sekkei_data_id(K->J):-
    retract( sekkei_data_id(K)),
    J is K + 1,
    assert(sekkei_data_id(J)).

jido_kikan_sekkei(M):-
    init_sekkei_data,
    kabusiki_gaisha_no_kikan_sekkei(X, [A,B|Y]),
    update_sekkei_data_id(_->J),
    assert( sekkei_data( J,X, [A,B|Y])),
    print_sekkei(mode(M), J, X, [A,B|Y]),
    fail.

print_sekkei(mode(r), J,_, [A,B|Y]):-
    findall(Q, (member(S,Y), (S=yes->Q=o;Q='-')),L),
    nl,write(J:[A,B|L]).

```

図 4. PROLOG によって論理的に可能な機関設計を生成するためのルール

2. PROLOG による機関設計原則のモデリング

株式会社の機関設計ルール（図 1 参照）を，PROLOG による論理的モデルに翻訳すれば，例えば図 2 および図 3 のようになる．表 1 にはそのプログラムを構成する節形式をまとめた．² これらのルールの基本構造は，そうむずかしくない．まず株主総会と取締役はすべてのパターンで必須である．株式会社は公開会社と株式譲渡制限会社に分かれるが，公開会社では取締役会設置が義務である．また大会社の場合は，会計監査人を設ける義務がある．任意設置の場合を含めて，取締役会や会計監査人を設けるためには，チェック機関を適切に選んで設けないといけないが，その制約条件がやや込み入っており，そのためいくつものパターンに分岐する．

² 図 1 に示す原則は，実際の会社法の条文とは異なるが，論理的な意味は等しい(第 5 節参照)．

3. 機関設計の論理的に可能なパターンを生成する

本節では、前節で作成した株式会社の機関設計の PROLOG コードを用いて、これを実験で確かめてみよう。株式会社の機関設計は、文献[2]や監査会社のウェブサイトなどで述べられていることが正しければ、全部で 39 パターン³あるはずである。図 4 に示す実験用スクリプト jido_kikan_sekkei/1 は、前節のコードを利用して、論理的に可能な機関設計のパターンを（番号付で）生成し、表示する。その他、図 4 中には、実験の支援や結果の表示のためのコードが若干追加されている。実験結果は事実節 sekkei_data/3 として記録される。図 5 はその実験結果のデータを Excel に取り込んで整えたものである。

表 3. 可能な機関設計の集計: (a) 全パターン, (b) 取締役会を設置する場合, (c) 会計監査人を設定する場合, (d) 取締役会と会計監査人をともに設置する場合

(a)

取締役会	(複数のアイテム)
会計監査人	(複数のアイテム)

データの個数 / パターン	会社規模		
会社種類	大会社	中小会社	総計
株式譲渡制限会社	8	17	25
公開会社	4	10	14
総計	12	27	39

(c)

取締役会	(複数のアイテム)
会計監査人	o

データの個数 / パターン	会社規模		
会社種類	大会社	中小会社	総計
株式譲渡制限会社	8	8	16
公開会社	4	6	10
総計	12	14	26

(b)

取締役会	o
会計監査人	(複数のアイテム)

データの個数 / パターン	会社規模		
会社種類	大会社	中小会社	総計
株式譲渡制限会社	6	11	17
公開会社	4	10	14
総計	10	21	31

(d)

取締役会	o
会計監査人	o

データの個数 / パターン	会社規模		
会社種類	大会社	中小会社	総計
株式譲渡制限会社	6	6	12
公開会社	4	6	10
総計	10	12	22

³ 付録 2 に示すように、会社の種類と規模を除く機関設計の基本パターンは、17 通り(大会社では 8 通り)である。

		機関番号:	1	2	3	4	5	6	7	8
パタンNo.	会社種類	会社規模	取締役	取締役会	監査役	監査役会	三委員会	会計参与	会計監査人	株主総会
1	公開会社	大会社	○	○	○	○	-	○	○	○
2	公開会社	大会社	○	○	○	○	-	-	○	○
3	公開会社	大会社	○	○	-	-	○	○	○	○
4	公開会社	大会社	○	○	-	-	○	-	○	○
5	公開会社	中小会社	○	○	○	○	-	○	○	○
6	公開会社	中小会社	○	○	○	○	-	○	-	○
7	公開会社	中小会社	○	○	○	-	-	○	○	○
8	公開会社	中小会社	○	○	○	-	-	○	-	○
9	公開会社	中小会社	○	○	○	○	-	-	○	○
10	公開会社	中小会社	○	○	○	○	-	-	-	○
11	公開会社	中小会社	○	○	○	-	-	-	○	○
12	公開会社	中小会社	○	○	○	-	-	-	-	○
13	公開会社	中小会社	○	○	-	-	○	○	○	○
14	公開会社	中小会社	○	○	-	-	○	-	○	○
15	株式譲渡制限会社	大会社	○	○	○	○	-	○	○	○
16	株式譲渡制限会社	大会社	○	○	○	-	-	○	○	○
17	株式譲渡制限会社	大会社	○	○	○	○	-	-	○	○
18	株式譲渡制限会社	大会社	○	○	○	-	-	-	○	○
19	株式譲渡制限会社	大会社	○	○	-	-	○	○	○	○
20	株式譲渡制限会社	大会社	○	○	-	-	○	-	○	○
21	株式譲渡制限会社	大会社	○	-	○	-	-	○	○	○
22	株式譲渡制限会社	大会社	○	-	○	-	-	-	○	○
23	株式譲渡制限会社	中小会社	○	○	○	○	-	○	○	○
24	株式譲渡制限会社	中小会社	○	○	○	○	-	○	-	○
25	株式譲渡制限会社	中小会社	○	○	○	-	-	○	○	○
26	株式譲渡制限会社	中小会社	○	○	○	-	-	○	-	○
27	株式譲渡制限会社	中小会社	○	○	○	○	-	-	○	○
28	株式譲渡制限会社	中小会社	○	○	○	○	-	-	-	○
29	株式譲渡制限会社	中小会社	○	○	○	-	-	-	○	○
30	株式譲渡制限会社	中小会社	○	○	○	-	-	-	-	○
31	株式譲渡制限会社	中小会社	○	○	-	-	○	○	○	○
32	株式譲渡制限会社	中小会社	○	○	-	-	○	-	○	○
33	株式譲渡制限会社	中小会社	○	○	-	-	-	○	-	○
34	株式譲渡制限会社	中小会社	○	-	○	-	-	○	○	○
35	株式譲渡制限会社	中小会社	○	-	○	-	-	○	-	○
36	株式譲渡制限会社	中小会社	○	-	○	-	-	-	○	○
37	株式譲渡制限会社	中小会社	○	-	○	-	-	-	-	○
38	株式譲渡制限会社	中小会社	○	-	-	-	-	○	-	○
39	株式譲渡制限会社	中小会社	○	-	-	-	-	-	-	○

図 5. 実験によって生成された機関設計の 39 パタン

表 3(a)～(d)はピボットテーブルによる実験データの集計である。例えば表 3(a)は全パターン数 39 通りの内訳を示す。それぞれ大会社が 12 通り, 中小会社が 27 通りあることや, 公開会社が 14 通り, 株式譲渡会社が 25 通り, また両者のクロス集計では, 例えば大会社の公開会社が 4 通り, 大会社の株式譲渡制限会社 8 通りある。表 3(a)は, 取締役会を設置する 31 の可能なパターンを抽出し, 集計している。設置義務がある公開会社については表 1(a)と変わらないが, 非公開の中小会社では 7 つ減って 11 通りになっている。

4. 最適な機関設計を選択する

前節の実験によって, 第 2 節で論じた PROLOG コード (図 2 および図 3 参照) は, それを過不足なく生成することが分かった。本節では, 機関設計の最適選択について考えよう。まず, 機関設計の代替案の集合に対するプリファレンスを, 創業者あるいは親会社の立場から, モデリングする必要がある。可能な機関設計案は, 何らかの選択基準に照らして評価・吟味され, 最もよいものが推奨される。最終的には, 彼らが納得あるいは合意するものが選ばれる。

プリファレンスモデルの方法としては, 基準値(期待効用, 費用対効果比, 多基準効用など), 優先順序, あるいは論理的な制約条件などがある。また, 意思決定論や OR のテキストからは, さまざまな選択基準を学ぶことができる。単純なスカラー値(利益や費用)の最大化ないし最小化, 多基準の場合のマクシミン基準, ハービッツ基準, ベイズ基準, ミニマックスリグレット基準など。また制約条件つき問題解決のお手本である, 線形計画法や目標計画法による定式化も考えられる。さらにはゲーム理論や社会的選択理論によるモデリングが適するかもしれない。なぜならば機関設計の複雑さは, いてみれば経営と所有の分離によって発生するエージェンシー問題を防止するための代償であるから。しかし, これら古典的な, あるいはアドバンスドな代替的モデリングに, 本論文ではこれ以上立ち入らないことにする。

```

select_sekkei((P_list,N_list), J,X,[A,B|Y]):-
    (var(P_list)->P_list=[1,8];true),
    (var(N_list)->N_list=[];true),
    sekkei_data(J,X,[A,B|Y]),
    \+ (member(X, P_list), \+ is_selected(X,_,Y)),
    \+ (member(X, N_list), is_selected(X,_,Y)).

is_selected(X,(Kikan,N),Y):-
    nth1(N,Y,yes),
    kikan(N,Kikan),
    member(X,[N,Kikan]).

kikan(1,torisimari-yaku).
kikan(2,torisimari-yaku-kai).
kikan(3,kansa-yaku).
kikan(4,kansa-yaku-kai).
kikan(5,san-iinkai).
kikan(6,kaikei-sanyo).
kikan(7,kaikei-kansa-nin).
kikan(8,kabunusi-sokai).

```

図 6. ポジティブリストとネガティブリストを用いた機関設計選択ルール

```

?- A=joto-seigen,B=chu-sho,P_list=[1],N_list=[2,7],
select_sekkei((P_list,N_list), J,X,[A,B|Y]),
print_sekkei(mode(_, [1,8]),J,X,[A,B|Y]),fail.

35:[株式譲渡制限会社, 中小会社, 監査役, 会計参与]
37:[株式譲渡制限会社, 中小会社, 監査役]
38:[株式譲渡制限会社, 中小会社, 会計参与]
39:[株式譲渡制限会社, 中小会社]

No
?-

```

図 7. リストを用いた機関設計選択ルールの実行例：公開会社でない中小会社で取締役会(2)と会計監査人(7)のいずれもおかない場合

本論文では、これらに代わるより簡便かつ柔軟な方法として、2種類の指定リストを用いて創業者のプリファレンスとその選択基準を扱う。すなわち望ましい機関設定に含まれるべき機関の集合（ポジティブリスト；P-List）と含みたくない機関の集合（ネガティブリスト；N-List）を、機関は種類または番号で、それぞれのリストによって指定する。この方法を実現するためには、図6に示すような実験データの抽出ルールを追加するだけでよい。⁴

図7は、機関リストの2番と7番がネガティブリストに指定されたときの、図6のコードの実行例を示している。コマンドプロンプト?-の後に、3行にわたって入力されているのが、そのゴール節である。また同様の工夫は、表示ルーチン print_sekkei/4 に対しても適用できる。図7の出力表示では、株主総会(8)と取締役(1)（これらはすべてのパターンに含まれる）が省かれていることに注意する。日本語による表示については、次節で説明する。

費用などの観点から、より複雑でない機関設計が優先される場合、（リストの短さに費用が比例すると仮定して）図7の39番の設計案、つまり取締役と株主総会のみが最適である。より一般的には、機関設計の複雑性は、機関ごと、あるいは複数の機関の組み合わせ方によって異なるだろう。⁵ コードの拡張については読者の練習問題とする。

5. 自然言語による表示と法令文の論理的モデル

まず、機関設計の支援を自然言語によって行うための基礎について、簡単に触れておきたい。商用の PROLOG システムは、たいてい日本語に対応しているが、そうでない場合でも、日本語表記（あるいは文字コード自体）の辞書を追加することにより、ある程度対応できる。⁶ 図8に前節の実行例（図7）で用いた辞書などを示す。⁷

⁴ あるいは論理演算によるデータベースの検索ができるもの、例えば、付録2に示すピボットテーブル(Excel)でもよい。

⁵ 機関設計は、より広い文脈では企業統治の一部である。すなわち、(1)迅速な意思決定を可能にするため、より簡略な機関を選ぶことと、(2)経営の規律付け・監視のための機関を設けることは、ともに企業価値・株主重視を実現するという上位目標を実現するための下位目標であるが、それらのバランスが、企業規模や発行株式の種類に応じて必要となる。

⁶ 例えば、フリーソフトの SWI-PROLOG(5.0.10)では、日本語フォントセットを選べば、引用符で囲った日本語文字列を write/1 で表示できる。

⁷ 法令翻訳データ集(<http://www.cas.go.jp/jp/seisaku/hourei/data1.html>)を参考にしたが、会計参与など対訳不明である。

```

dictionary('株式会社', kabusiki-kaisha, business_corporation).
dictionary('公開会社', kokai-kaisha, business_corporation_1).
dictionary('株式譲渡制限会社', kabusiki-joto-seigen-kaisha, business_corporation_2).

dictionary('株主総会', kabunusi-sokai, shareholders_meeting).
dictionary('取締役', torisimari-yaku, director).
dictionary('取締役会', torisimari-yaku-kai, board_of_directors).
dictionary('監査役', kansa-yaku, corporate_auditor).
dictionary('監査役会', kansa-yaku-kai, board_of_corporate_auditors).
dictionary('三委員会', san-iinkai, three_committees).
dictionary('指名委員会', simeiiinkai, nominating_committee).
dictionary('報酬委員会', hoshu-iinkai, compensation_committee).
dictionary('監査委員会', kansa-iinkai, audit_committee).
dictionary('会計参与', kaikei-sanyo, '?').
dictionary('会計監査人', kaikei-kansa-nin, accounting_auditor).

print_sekkei(mode(j), J, (A,B), [_,_|Y]) :-
    findall(JName, (
        nth1(N,Y, yes),
        kikan(N,Kikan),
        dictionary(JName,Kikan,_)
    ), L),
    nl, write(J:[A,B|L]).

```

図 8. 機関名の日本語表記辞書とその表示用ルール

ところで、法令の条文、条文を一般向けに分かりやすく示したその翻案、および論理的モデル（許可される機関設計パターン）という 3 つ組みは、実用上、セットで用いられる。もし翻案と条文の論理的な意味が等しくなければ、（すなわち許される機関設計の必要十分条件でないならば、）誤りである。あるいは、冗長なルールが含まれている場合は、矛盾はしないが、実用上メリットがないのなら削除した方がより簡潔になるだろう。

実を言うと、第 1 節の図 1 に示した監査法規委員会[2]による 8 原則と、会社法[1]における条文とは、若干、というか一見するとかなり、違っている。株主総会以外の機関を定めた、実際の条文は図 9 のようである。しかしながら、翻案（図 1）の正当性は、例えば図 5 の Excel シート（とオートフィルター）によって、確かめることができる。

あるいは、付録 1 に示した検証用コードによって、その十分性を示すことができる（実験結果は図 10 を参照）。一方、もし不正確な翻案であれば、その PROLOG コードを試すと、とうぜ

んながら、図 5 のパターンと一致しない。その場合、このコードは反例を表示する。

例えば、図 2 中の監査役会のルール kansa-yaku-kai/2 内にある sekkei_rule の 327-4 を無効化（つまりプログラムの行頭に%を付けてコメントアウト）しても、39 パターンは変わらない。つまり、図 1 の④における括弧書きは、論理的には冗長である。数学的に言えば、残りのルールを公理としたときの、定理の一つである。もちろん、定理はたとえトートロジーであっても、人間の理解力の限界を補うならば、それなりに価値があるといえるだろう。そうでなければ、削ってもよいはずである。

一方、図 1 の②のただしがきの部分に相当する PROLOG コード、すなわち図 3 で sekkei_rule の 327 の 2 つ目のルール節を無効化すると、1 パターン減って 38 パターンになる。このときの PROLOG モデルは、条文に対して十分性を満たすが、必要性を満たさない。

- | |
|--|
| <ul style="list-style-type: none">i. 株式会社には、1 人又は 2 人以上の取締役を置かねばならない。（326 条 1 項）ii. 株式会社は、定款の定めによって、取締役会、会計参与、監査役、監査役会、会計監査人又は委員会を任意に置くことができる。（326 条 2 項）iii. 次に掲げる株式会社は、取締役会を設置しなければならない。（327 条 1 項）<ul style="list-style-type: none">1. 公開会社2. 監査役会設置会社3. 委員会設置会社iv. 取締役会設置会社（委員会設置会社を除く）は、監査役を設置しなければならない。ただし、公開会社でない会計参与設置会社については、この限りではない。（327 条 2 項）v. 会計監査人設置会社（委員会設置会社を除く）は、監査役を設置しなければならない。（327 条 3 項）vi. 委員会設置会社は、監査役を設置してはならない。（327 条 4 項）vii. 委員会設置会社は、会計監査人を設置しなければならない。（327 条 5 項）viii. 大会社（公開会社でないもの及び委員会設置会社を除く。）は、監査役会および会計監査人を設置しなければならない。（328 条 1 項）ix. 公開会社でない大会社は、会計監査人を設置しなければならない。（328 条 2 項） |
|--|

図 9. 機関設計についての実際の条文

```

?- [kaisha].
% kaisha compiled 0.00 sec, 0 bytes

Yes
?- jido_kikan_sekkei(r).

1:[kokai, dai, o, o, o, o, -, o, o, o]
2:[kokai, dai, o, o, o, o, -, -, o, o]
3:[kokai, dai, o, o, -, -, o, o, o, o]
4:[kokai, dai, o, o, -, -, o, -, o, o]
5  . . .
35:[joto-seigen, chu-sho, o, -, o, -, -, o, -, o]
36:[joto-seigen, chu-sho, o, -, o, -, -, -, o, o]
37:[joto-seigen, chu-sho, o, -, o, -, -, -, -, o]
38:[joto-seigen, chu-sho, o, -, -, -, -, -, o, -, o]
39:[joto-seigen, chu-sho, o, -, -, -, -, -, -, -, o]

No
?- kaisha_ho(_,_),fail.

No
?-

```

図 10. 条文に対する翻案の正当性（充分性）を検証した

```

?- jido_kikan_sekkei(r).

1:[kokai, dai, o, o, o, o, -, o, o, o]
2:[kokai, dai, o, o, o, -, -, o, o, o]
3:[kokai, dai, o, o, o, o, -, -, o, o]
4:[kokai, dai, o, o, o, -, -, -, o, o]
5:[kokai, dai, o, o, -, -, o, o, o, o]
6:[kokai, dai, o, o, -, -, o, -, o, o]
7:[kokai, chu-sho, o, o, o, o, -, o, o, o]
. . .
36:[joto-seigen, chu-sho, o, -, o, -, -, o, -, o]
37:[joto-seigen, chu-sho, o, -, o, -, -, -, o, o]
38:[joto-seigen, chu-sho, o, -, o, -, -, -, -, o]
39:[joto-seigen, chu-sho, o, -, -, -, -, -, o, -, o]
40:[joto-seigen, chu-sho, o, -, -, -, -, -, -, -, o]

No
?- kaisha_ho(viii,328-1).

violation: (viii, 公開会社, 大会社)
2:[公開会社, 大会社, 取締役会, 監査役, 会計参与, 会計監査人]

No
?-

```

図 11. 一部のルールを無効化し、条文に対する違反を検出した

さらに、会計監査人をおくための要件を述べた図 1 の⑥において、大会社であって譲渡制限会社でない株式会社についての括弧書きに注目しよう。図 2 のコード中では `kaikai-kansanin/2` というルール節に対応するが、その本体にある `sekkei_rule` の 327-3-5 の部分を無効化すると、今度は 2 パタン増えて全部で 40 パタン出力される。この 2 パタンは公開会社の大会社で取締役会を置くが、監査役会を置かない。この部分是对应する条文がやや分かりにくく、327 条 3 項ではなく、328 条 1 項（図 9 では条文番号 viii）に違反する。実際、前出の検証用コード（付録 1）を用いると、図 11 に示すように違反パタンと対応する条文番号が検出された。

6. おわりに

本論文では、論理プログラミング言語 PROLOG を用いて、会社法によって認められた株式会社の機関設計のルールを表現し、またそのすべての可能な機関設計パタン 39 通りをその PROLOG コードによって生成した。また、日本語による表示を含めて、指定の条件に合った機関設計を選択する方法を論じた。なお、法令の条文（ないしその翻案）から PROLOG コードを生成すること、あるいはその逆は、第 5 節で試みたことの自然な延長であるけれども、これについては機会を改めたい。

参考文献

- [1] 法務省民事局: 「会社法」の概要. [URL: <http://www.moj.go.jp/HOUAN/houan33.html#03>]
- [2] 日本監査役協会監査法規委員会: 監査役からみた会社法の捉え方, 監査法規委員会報告, 2005. [URL: <http://www.kansa.or.jp>]
- [3] Clocksin, W. F. and Mellish, C. S.: *Programming in Prolog: Using ISO Standard*, 5th edition, Springer Verlag, 2003.

付録 1. 図 9 の条文に対する違反を検出するプログラム

<pre> kaisha_ho(iii,327-1):- kikan(K,torisimari-yaku-kai), kikan(P,kansa-yaku-kai), kikan(Q,san-iinkai), \+ (select_sekkei(([K]), J,X,[A,B Y]), (A=kokai;nth1(P,Y,yes);nth1(Q,Y,yes)) ,write_violation(iii,J,X,[A,B Y])). kaisha_ho(iv,327-2):- kikan(K,torisimari-yaku-kai), kikan(P,san-iinkai), kikan(F,kansa-yaku), kikan(G,kaikei-sanyo), \+ (select_sekkei(([K],[F,P]), J,X,[A,B Y]), \+ (A \= kokai, nth1(G,Y,yes)) ,write_violation(iv,J,X,[A,B Y])). kaisha_ho(v,327-3):- kikan(O,kaikei-kansa-nin), kikan(P,san-iinkai), kikan(F,kansa-yaku), \+ (select_sekkei(([O],[F,P]), J,X,[A,B Y]) ,write_violation(v,J,X,[A,B Y])). kaisha_ho(vi,327-4):- kikan(P,san-iinkai), kikan(F,kansa-yaku), \+ (select_sekkei(([F,P],[]), J,X,[A,B Y]) ,write_violation(vi,J,X,[A,B Y])). </pre>	<pre> kaisha_ho(vii,327-5):- kikan(O,kaikei-kansa-nin), kikan(P,san-iinkai), \+ (select_sekkei(([P],[O]), J,X,[A,B Y]) ,write_violation(vii,J,X,[A,B Y])). kaisha_ho(viii,328-1):- B=dai, kikan(F,kansa-yaku-kai), kikan(O,kaikei-kansa-nin), kikan(P,san-iinkai), \+ (select_sekkei(([P]), J,X,[A,B Y]), B = kokai, \+ (nth1(F,Y,yes),nth1(O,Y,yes)) ,write_violation(viii,J,X,[A,B Y])). kaisha_ho(ix,328-2):- B=dai, kikan(O,kaikei-kansa-nin), \+ (select_sekkei(([O]), J,X,[A,B Y]), B \= kokai ,write_violation(ix,J,X,[A,B Y])). write_violation(R,J,X,[A,B Y]):- nl, write(violation:(R,X)), print_sekkei(mode([1,8]),J,X,[A,B Y]). </pre>
--	---

付録 2. 17 の基本パターン(上), 大会社の 8 パターン (中), および図 7 との比較(下). ただし株主総会と取締役会を表示していない.

torishimari_yaku_kai	(すべて)
kansa_yaku	(すべて)
kansa_yaku_kai	(すべて)
san_iinkai	(すべて)
kaikei_sanyo	(すべて)
kaikei_kansa_nin	(すべて)

データの個数 / id	kaisha		kibo		総計
	株式譲渡制限会社		公開会社		
	大会社	中小会社	大会社	中小会社	
取締役会+ 監査役+ 監査役会++ 会計参与+会計監査人	1	1	1	1	4
取締役会+ 監査役+ 監査役会+++ 会計監査人	1	1	1	1	4
取締役会++++ 三委員会+ 会計参与+会計監査人	1	1	1	1	4
取締役会++++ 三委員会++ 会計監査人	1	1	1	1	4
取締役会+ 監査役+ 監査役会++ 会計参与+-		1		1	2
取締役会+ 監査役+ 監査役会+++ 会計参与+会計監査人	1	1		1	3
取締役会+ 監査役+ 監査役会+++ 会計参与+-		1		1	2
取締役会+ 監査役+ 監査役会++++		1		1	2
取締役会+ 監査役++++ 会計監査人	1	1		1	3
取締役会+ 監査役+++++		1		1	2
+ 監査役++++ 会計参与+会計監査人	1	1			2
+ 監査役+++++ 会計監査人	1	1			2
取締役会+++++ 会計参与+-		1			1
+ 監査役++++ 会計参与+-		1			1
+ 監査役+++++		1			1
+++++ 会計参与+-		1			1
++++++		1			1
総計	8	17	4	10	39

kibo	大会社
torishimari_yaku_kai	(すべて)
kansa_yaku	(すべて)
kansa_yaku_kai	(すべて)
san_iinkai	(すべて)
kaikei_sanyo	(すべて)
kaikei_kansa_nin	(すべて)

データの個数 / id	kaisha		総計
	株式譲渡制限会社	公開会社	
取締役会+ 監査役+ 監査役会++ 会計参与+会計監査人	1	1	2
取締役会+ 監査役+ 監査役会+++ 会計監査人	1	1	2
取締役会++++ 三委員会+ 会計参与+会計監査人	1	1	2
取締役会++++ 三委員会++ 会計監査人	1	1	2
取締役会+ 監査役+ 監査役会+++ 会計参与+会計監査人	1		1
取締役会+ 監査役++++ 会計監査人	1		1
+ 監査役++++ 会計参与+会計監査人	1		1
+ 監査役+++++ 会計監査人	1		1
総計	8	4	12

kibo	中小会社
torishimari_yaku_kai	-
kansa_yaku	(すべて)
kansa_yaku_kai	(すべて)
san_iinkai	(すべて)
kaikei_sanyo	(すべて)
kaikei_kansa_nin	-

データの個数 / id	kaisha	
	株式譲渡制限会社	総計
+ 監査役++++ 会計参与+-	1	1
+ 監査役+++++	1	1
+++++ 会計参与+-	1	1
++++++	1	1
総計	4	4