『論理と形式化』確認課題 (期末レポート) の解答例

亀山幸義 (kam@cs.tsukuba.ac.jp)

確認課題 (期末レポート) について，その解答の一例，あるいは解説を掲載する．自分が解いた問題について，各自で答え合わせをして，質問・疑問があれば，教員あてメイル等で連絡してほしい．

1 第 7 回課題の演習問題 4

問題: 「A が証明可能」と「$\neg A$ が充足可能でない」が同値であることを示せ．

解答例:

(左から右への証明) A が証明可能と仮定する．命題論理の健全性より A は恒真 (妥当,valid) である．つまり，A は全ての真理値割当てのもとで真である．よって，$\neg A$ は全ての真理値割当てのもとで偽であり，充足可能でない．

(右から左への証明) $\neg A$ が充足可能でないと仮定する．$\neg A$ は全ての真理値割当てのもとで偽である．よって，A は全ての真理値割当てのもとで真であり，恒真 (妥当) である．命題論理の完全性より，A は証明可能である．

また,「A が充足可能」と「$\neg A$ が証明可能でない」の同値性は、上と同様に証明してもよいし、上の事実で A を $\neg A$ にしたものを考えても示せる。(ここでは省略)

2 第 7 回演習問題 5 の直前の「課題」

問題: Tseitin 変換 ϕ は，ある定数 C_1,C_2 に対して，任意の命題論理式 A に対して $|\phi(A)| < C_1|A| + C_2$ を満たすことを示せ．

解答例: テキスト No.7 で与えた形であれば，$C_1 = 12$,$C_2 = 2$ と置けばよい．そのことの証明の概要は以下の通り．

テキスト p.16 にある「最後に $\wedge X_n$ をつける」というステップをやる前の Teisein 変換を ϕ' として，$|\phi'(A)| \leq 12 \cdot |A|$ を示そう．A の構造に関する帰納法で証明する。

(Base Case: A が原子命題) この場合，$|\phi'(A)| = 0$ かつ $|A| = 0$ なので，上記不等式は成立する．

(Step: $A = B \wedge C$ のとき) この場合, $\phi'(A)$ は，テキスト p.15 の $X_1 \equiv P \wedge Q$ の右側に書いてある CNF(を適当に名前換えしたもの) と，$\phi'(B)$ および $\phi'(C)$ を $\wedge$ でつなげた論理式になる．その CNF の論理記号は 10 個であるので,

$$|\phi'(A)| = 10 + (|\phi'(B)| + 1) + (|\phi'(C)| + 1)$$

である．(ここで $+1$ しているのは，CNF をつなげるための $\wedge$ をカウントしているからである．) ここで、帰納法の仮定を B, C に適用すると、$|\phi'(B)| \leq 12 \cdot |B|$ かつ $|\phi'(C)| \leq 12 \cdot |C|$ である。よって、

$$\begin{aligned} |\phi'(A)| &\leq 12 + 12 \cdot |B| + 12 \cdot |C| \\ &= 12 \cdot (|B| + |C| + 1) \\ &= 12 \cdot |A| \end{aligned}$$

よって証明できた。

(Step: $A = B \vee C$, $A = \neg B$, $A = B \supset C$ のとき) 上と同様の計算である．(これら 3 つのケースでは、$B \wedge C$ のときよりサイズが小さい式になる。)

以上から，$|\phi(A)| = |\phi'(A)| + 1 < 12 \cdot |A| + 2$ である．(最後のステップで $\wedge X_n$ を追加するので 1 を加算した。)

3 第 7 回演習問題 8

前半の SAT ソルバにかける問題は、実際に「やる」だけなので、解答は省略する。(ちゃんと「やった」ことを証明するスクリーンショットなどをつけた答案を評価した。言葉だけでやった、とかいてあるものは減点した。)

後半の問題は、命題論理式として具体的に記述すればよいがいろいろな方式があるので、ここでは省略する。1 人だけ「自前の SAT ソルバ」を実装して解いた人がいて、素晴しかった。

4 第 10 回課題の演習問題その 1

型導出図を書く問題である。

$$\dfrac{\dfrac{}{y : \mathtt{int} \to \mathtt{int} \vdash y \ : \ \mathtt{int} \to \mathtt{int}} \quad \dfrac{}{y : \mathtt{int} \to \mathtt{int} \vdash 7 \ : \ \mathtt{int}}}{y : \mathtt{int} \to \mathtt{int} \vdash y \ 7 \ : \ \mathtt{int}}$$

次の導出図では、$\Gamma = y : \mathtt{int} \to \mathtt{int}$, $\Delta = y : \mathtt{int} \to \mathtt{int}, x : \mathtt{int}$ とする。

$$\dfrac{\dfrac{\dfrac{}{\Delta \vdash y \ : \ \mathtt{int} \to \mathtt{int}} \quad \dfrac{\dfrac{}{\Delta \vdash y \ : \ \mathtt{int} \to \mathtt{int}} \quad \dfrac{}{\Delta \vdash x \ : \ \mathtt{int}}}{\Delta \vdash y \ x \ : \ \mathtt{int}}}{\Delta \vdash y \ (y \ x) \ : \ \mathtt{int}}}{\Gamma \vdash \lambda x.(y \ (y \ x)) \ : \ \mathtt{int} \to \mathtt{int}}$$

次の導出図では、$\Gamma = y : \mathtt{int} \to \mathtt{int}$, $\Delta = y : \mathtt{int} \to \mathtt{int}, x : \mathtt{int}$, $\Sigma = y : \mathtt{int} \to \mathtt{int}, x : \mathtt{int}, y : \mathtt{int}$ とする。

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Sigma \vdash x \ : \ \mathtt{int}} \quad \dfrac{}{\Sigma \vdash y \ : \ \mathtt{int}}}{\Sigma \vdash x + y \ : \ \mathtt{int}} \quad \dfrac{}{\Sigma \vdash x \ : \ \mathtt{int}}}{\Sigma \vdash (x + y) + x \ : \ \mathtt{int}}}{\Delta \vdash \lambda y.(x + y) + x \ : \ \mathtt{int} \to \mathtt{int}}}{\Gamma \vdash \lambda x.\lambda y.(x + y) + x \ : \ \mathtt{int} \to (\mathtt{int} \to \mathtt{int})}$$

5 Prolog プログラミング

以下のプログラムを Prolog で書きなさい．ただし、自然数は 0,$s(0)$ などで表現されるものとする。

- 階乗をあらわす述語 $Factorial(X, Y)$.
- 2 つの自然数の最大公約数をあらわす述語 $GCD(X, Y, Z)$.
- 2 つの自然数の最小公倍数をあらわす述語 $LCM(X, Y, Z)$.
- 素数かどうかをあらわす述語 $Prime(X)$.

(問題文に明記してある通り，レポートには，プログラムを書くだけでなく，プログラムの説明と複数の実行例を書く必要があるが，ここでは省略する．)

Factorial:

```
add(0,Y,Y).
add(s(X),Y,s(Z)) :- add(X,Y,Z).
times(0,Y,0).
times(s(X),Y,Z) :- times(X,Y,W), add(W,Y,Z).
factorial(0,s(0)).
factorial(s(X),Z) :- factorial(X,Y),times(Y,s(X),Z).
```

GCD:

```
lt(0,s(X)).
lt(s(X),s(Y)) :- lt(X,Y).
geq(X,0).
geq(s(X),s(Y)) :- geq(X,Y).
sub(X,0,X).
sub(s(X),s(Y),Z) :- sub(X,Y,Z).
gcd(X,0,X).
gcd(0,X,X).
gcd(X,Y,Z) :- geq(X,Y),sub(X,Y,W),gcd(W,Y,Z).
gcd(X,Y,Z) :- lt(X,Y),sub(Y,X,W),gcd(W,X,Z).
```

lt は less-than を意味する．geq は greater-than-or-equal を意味する．sub は引き算である。

LCM:

```
div(X,Y,0) :- lt(X,Y).
div(X,Y,s(Z)) :- geq(X,Y),sub(X,Y,W),div(W,Y,Z).
lcm(X,Y,Z) :- gcd(X,Y,W),div(X,W,V),times(V,Y,Z).
```

div は、正の整数上の割り算 (余りを切り捨て) である。

Prime:

```
nonmul(s(X),Y) :- lt(s(X),Y).
nonmul(s(X),Y) :- geq(s(X),Y),sub(s(X),Y,Z),nonmul(Z,Y).
prime2(X,s(0)).
prime2(X,s(Y)) :- nonmul(X,s(Y)),prime2(X,Y).
prime(s(X)) :- prime2(s(X),X).
```

nomul(X,Y) は、X が Y の倍数でないとき真となる。prime2(X,Y) は、X が 2 以上 Y 以下のすべての整数で割り切れないとき真になる。すると、prime(X) は、意味的には、prime2(X,X-1) という感じとなり、1 を引き算するかわりに上記のように表現した。

補足: 言うまでもなくプログラムの書き方は多様であり、上記以外にも正解は多数ある。

採点にあたっては、プログラムだけでなく、プログラムの説明があるかどうか、また、複数の実行例を書いてあるかどうか (5 個程度あれば完璧である) を採点した。また、自然数を $0, s(0), \cdots$ で表現せよ、とかいてあるのに $0, 1, 2, \cdots$ で表現している答案は、評価できなかった。

プログラムのテストが甘くて、入力の値によっては無限ループになってしまう答案などもあったが、そもそも Prolog であるために無限ループはさけがたいところもあり、(正しく動く場合だけちゃんとしていれば) 減点しなかった。