

「情報」試験攻略メモ(シケプリ)

2010 年度入学 理 1-23 組

*田中哲郎氏の授業の人専用です。他の教員の授業の人はあまり参考にならないと思います。

[1] 出題範囲と対策

1. 共通問題

出題範囲は以下のURLを参照。

<http://www.edu.c.u-tokyo.ac.jp/edu/requisites2010.htm>

また、過去問の傾向をみると、共通問題はほとんどが教科書に書いてある内容を丸暗記していれば解けてしまう問題が多いものの、専門用語を覚えていないと全く話にならない。

そのため、専門用語とその定義、そしてそれらの関係を抑えることが必須。授業用スライドで十分説明されているので、本来ならそれを読むだけで本来は事足りるが、共通問題部分において授業用スライドと同じような内容をまとめた他のクラスのシケプリがあったので、別に用意しておいた。

よく出題されるのは単語(先述シケプリの太字部分)の意味を理解しているかを確認するための問題なので、専門的な単語を覚えて理解することに専念した方がいい。特に出題率が高いのは、3章の内容で、5,7,10章も割と出題されている。他の章もそこまで頻度が高いとは言えないが、出題されないと言い切ることはできない。また10章に関する問題は社会とのつながりの強い問題で、選択式となっている事が多い。専門的な用語や発想の必要性が少なく済むため、下手にマニアックな問題を選ぶよりも社会的な問題を選ぶ方が簡単かもしれない。

2. 独自問題

独自問題は多くは過去の演習の内容に基づき、共通問題にて触れられていない内容が中心となっている。過去問の傾向からみる限り、オートマトン、情報量の計算、コンピュータ言語(ED21)、IC ゲートに関して対策を行えば大体事足りる内容である。

しかし、授業の演習が何も詰まることなくできた人ならば対策については全く必要性がなく、授業でやった事が理解できなかった人にとっては何をどう対策すれば点が来るかは分からない。どうしても専門的な発想力が必要となるため、初見でも完答できる人と、いくら対策をしても出来ない人に分かれるのは仕方ない。確実にできるようになることは保証できないが、ある程度できるように攻略の仕方をまとめることにした。

[2] 個別問題の対策

・オートマトン

どのように動いているか、という問題については難しくはない。適当に入力した場合を考えていけば簡単に見抜ける場合が多い。問題は「このプログラムの誤りを示せ・修正せよ」とう問題と、「ある処理を示すオートマトンを作成せよ」という問題である。

まず頻出問題として、電子錠のモデルがある。これはある形のコードを入力したら「必ず終了状態へ至る」プログラムが実態で、オートマトンの問題が出るたびに出ている。問題の内容はたいてい「ある入力をするれば必ず終了状態となるはずなのにならないのはなぜか？」のような問題であり、**原因は入力失敗時のジャンプ先の設定ミスであることがほとんどである。**

たとえば、“aabaa”という入力で終われば解錠とする場合、誤った瞬間にスタート地点に戻すとなると、“aaabaa”と入力したときに 3 番目の a の段階ではじめに戻されることに起因する。ここで、対策としては、**各状態を「既に有効な暗号として入力された文字列」が何かで分類し、最後の数文字の入力に注目しながら遷移を吟味しなすことがあげられる。**この例では、かっこ内がその文字列として、状態 0=(なし)、1=(a)、2=(aa)、3=(aab)、4=(aaba)、5=(aabaa)で、さらに状態 n では常に最後の n 字の入力に注意すると、0,1 からは a が押されれば 1,2、他は 0 へ、2 からは b が押されたら 3 だが、a が押された時は最後の二つは aa となるので、2 のままとなる。3 からは a が押されれば 4 で、b が押された時は b が二つ連続となっている部分が暗号に存在しないため 0 へ戻る。4 からは a が押されれば 5、他の場合は後ろの方が ~bab となり、このようなものの一部が最後の方となるものが暗号に含まれないため 0 に戻る。5 からは a が押された時は ~aaa という形で、最後から二つの aa までが有効な暗号と見れるため 2 へ、b が押された時は ~aab となり、最後の方がこの形と一致するのは 3 なので 3 へ移る。このように、後ろの入力に注目すれば、戻る先が始点とならないような暗号についても見抜くことができる。

ただし、練習を積んでおかないと本番いきなりやるのはかなり厳しい。過去問の類題や授業中の演習で扱ったものをやっておくことを推奨する。(2006,2008,2009 年過去問参照)

しかし、「必ず終了状態になる」入力の仕方と「必ず非終了状態になる」入力の仕方があるプログラムの場合、最短経路は同様に求められないし、プログラムのチェックの仕方も同様にはいかない。このようなとき、授業で扱った「自販機」の場合はいくら入れたかと、終了、すなわち購入状態か否か(購入状態にて残額は 0 か 50 である)で場合分けすることで、「残額 0 かつ非終了」、「残額 50 かつ非終了」、「残額 100 かつ非終了」、「残額 150 かつ非終了」、「残額 200 かつ非終了」、「残額 0 かつ終了」、「残額 50 かつ終了」の状態をどのような入力で移っていくかを考えればよかった。また、 m の倍数かどうかという問題は、 n 進数の数 a のとき、下の位に i がつくとき a の値が $an+i$ となる (2 進数では下の位に 0 が付くと 2 倍、1 が付くと 2 倍されて 1 が足された値となる)事と、 $a \equiv b \Rightarrow ak \equiv bk \pmod{\text{全ての自然数}}$ という合同式からも見える性質を利用して余りの値がどう遷移するかを考えることでオー

トマトンを作成できた。たとえば、2進数において3の倍数かどうかは、余りの値が、一番下の値にさらに0がつけば、元の値は2倍されるため、3で割ったときの余りが1であったものは2に(例：1,4,7 $\equiv$ 1は $\rightarrow$ 2,8,14 $\equiv$ 2(mod 3)に)、2であったものは1に(例：2,5,8 $\equiv$ 2 $\rightarrow$ 4,10,16 $\equiv$ 4 $\equiv$ 1(mod 3))、0であったものは0に(3の倍数を2倍しても3の倍数)なり、下の値にさらに1がつけば、元の値は2倍されて1が足されるため、3で割ったときの余りが1であったものが0に、2であったものは2に、0であったものは1となる。(数値は10進で示した)。しかしこのオートマトンの問題が出たらそれは点をただでもらったと思っていい。なぜならば、kで割った余りの種類は必ずk通りで、n進数において桁を増やしてもk通りのままであり続け、各状態からの遷移はn通りずつ、すなわち、**n進数のモデルでそれが何の倍数を示すかを答える問題の正解は n=各オートマトンから出ている分岐の数、k=オートマトンの数として、n進数の数をkで割ったときの余りがある値で終了状態となる。**(2008年過去問参照)

オートマトンが作成できるかどうかの条件は、2009年の過去問にそのような問題があったが、そこでの説明はいろいろとおかしいので(解答は、2i個だが、その前のbの数に触れていない等説明に問題がある)それなりに説明させてもらう。

オートマトンを作成できるかの判断は、「有限個の状態」で済むかである。2009年の過去問の説明をまともに書き換えると、

もし終了状態の判別が、 $2a \cdot b = 0$ であったとすると(a,b:a,bが入力された回数)、 $2a \cdot b$ の値によって各状態を説明できる。 $|2a \cdot b| \leq N$ とするNが存在したとすると、全体の状態の数は $2N+1$ という有限個であるが、 $2a \cdot b = N$ の状態からさらにaが押されれば $2a \cdot b = 2N+2$ となり $|2a \cdot b| \leq N$ に矛盾。ゆえに背理法の仮定は否定される。すなわち有限個の状態で示せない。

となる。

オートマトンの対策をまとめると、電子錠を作成する問題においてセルフチェックができるように、授業中にやった例題のオートマトンを一通り作れるように、オートマトン作成可能かどうかの判断ができるようにすること位である。これを超えるような内容の問題が出たらあきらめた方がいい。そのような問題は逸般人だけが解ければいい。

・情報量計算

情報量そのものの計算方法や意味は先述のシケプリを見ること。ただ平均情報量について述べられていなかったなのでここで定義式を書く。

(平均情報量) $:= \sum_i -p_i \log_2 p_i$ (p_i :状態 i である確率)

2007年の個別問題1は正直答えがよくわからなかったが、2009年の個別問題2では定義さえ押さえれば楽に解ける問題だった。これについては機械的にやるのみで、2007年の問題みたいなのは捨てる方針が正解なのでこれ以上述べない。

- ED21

これについては処理を目で追って行って、数値がどう変化するかを考えながら答えを出していくしかない。共通問題でも範囲となっているので、先述シケプリを参照とする。

- IC ゲート

これについても基本は先述シケプリを参照とするが、以下は覚えておかなければいけないものとして頭に入れておいてほしい。

- 加算器の構造
- 加減算器の構造
- RS・フリップフロップの構造
- D・FF またはレジスタの構造
- あるゲートの他のゲートでの表現(特に NAND での表現)

このあたりを押さえておけば試験で問題は生じないと思う。