2014年2月5日システムソフトウエア試験問題

学科　　学籍番号＿＿＿＿＿＿＿＿＿＿　　氏名＿＿＿＿＿＿＿＿＿＿

１．　つぎの文章の空白を埋めなさい．（各１点：合計50点）

アーキテクチャ・OS（26点）

- コンピュータは，CPU(中央処理装置)とメモリ，およびその他の入出力装置から構成されている．このうち，CPUは命令の読み取りとデコードなどを行う(A)　　部と，実際の計算を行う(B)　　部の２つから構成されている．(A)部には，演算対象や，読み取った命令と変数などを格納する(C)　　と呼ばれる小規模で高速なメモリが配置されている．
- CPUは命令の「(D)　　」と「(E)　　」を繰り返す．しかし，これでは(A)部と(B)部が交互にしか動かず，効率が良くない．これを解決するために，一つ前の命令の(E)と次の命令の(D)を同時に行うことによって高速化を実現する(F)　　処理が一般的に用いられている．
- (G)　　型のコンピュータは，メモリ上のプログラムを順次読み取り実行する．このために，次の命令読み取り位置を指しているのが，(H)　　という特殊な(C)である．
- 計算結果に応じて(H)の値を書き換える，つまり，実行するプログラムのメモリ上の位置を変えるのが(I)　　命令である．この命令を読み取った場合，前述の(F)処理で命令を読み取る場所が確定しないため，特別な機構を持たないCPUでは計算速度が低下する．
- 現代的なCPUの中には，制御部・演算部の組であるコアが複数備わった(J)　　，と呼ばれる構造のものがある．これらのCPUでは同一時刻に異なる命令が同時に実行できる．このような処理を(K)　　処理と呼ぶ．これに対して，複数のプログラムを短い時間間隔で順次コアに割り当てて実行する(L)　　処理により，各時刻では単一のプログラムしか実行していないが，ある程度の時間粒度で見ると複数のプログラムが同時に実行されているように見せることができる．これを並行処理と呼ぶ．
- 実行中のプログラムは，(M)　　と呼ばれ，OSからメモリやファイル，CPU割り当ての優先度などの(N)　　の割り当てを受ける．(M)はさらに(O)　　と呼ばれる単位に分けられて実行される．
- 補助記憶装置の一部を利用して，物理メモリの大きさよりも大きな論理メモリ空間を実現する手法を(P)　　と呼ぶ．この実現方式には，(M)が確保している物理メモリ単位で，補助記憶装置に退避・復帰させる(Q)　　と，「(R)　　」と呼ばれる固定サイズのメモリブロック単位で，補助記憶装置とメモリとの間で退避・復帰をさせる(S)　　の２種類がある．
- (S)では，物理メモリの空き容量が減った時点で補助記憶装置に不要なメモリを退避させておくという処理が必要になる．これを行うのが，(T)　　・デーモンである．このデーモンの動作を規定するアルゴリズムには，古い(R)の置き換え優先度を高くする(U)　　，最も長い期間使われていない(R)の置き換え優先度を高くする(V)　　などのアルゴリズムがある．

- ファイルは，補助記憶装置上のデータブロックをつなげた構造を持つ．このつなげ方は２種類あり，データブロック内に次のデータブロックを指すポインタを書く(W)構造と，データブロックを指すポインタの一次元並びを用いる(X)構造がある．
- HDD の場合，データブロックは HDD 上の(Y)に記録されたデータになる．
- (W)構造のファイルシステムの例として FAT がある．クラスタサイズ 16KB の FAT16 の場合，最大ファイルサイズは(Z)GB になる．

プログラム言語処理系（24 点）

- プログラム言語処理系は，C, C++, Pascal, Basic,などの(A)言語で書かれたプログラムテキストを，(1)CPU が解釈実行可能な(B)語に変換する(C)や，(2)逐次解釈・実行するインタープリタ，などで用いられる．
- プログラムテキストは，(D)文法で表現可能な字句の集まりとして表現可能であり，プログラムテキストからこれを抽出する字句解析のためにオートマトンが用いられる．
- 字句解析の後には，構文解析が行われる．通常は(E)文法を用いてプログラミング言語を定義することができる．
- 構文解析では，(F)記号に対してどのような順序で(G)が適用され，入力として与えられた終端記号の列が生成されたのかを分析する．この分析結果は(H)木で表すことができる．
- (H)木は，分岐のない中間節をまとめ，演算子を分枝節に引き上げることで(I)木（演算子木とも呼ばれる）に変換することができる．
- 算術式から得られた(I)木を(J)走査することにより(K)記法を生成し，これを入力としてスタック上で式の計算を行うことができる．
- 構文解析の方法は大きく分けて(L)型の構文解析と，(M)型の構文解析法がある．(L)型構文解析では最左導出を前提とした文法の下での構文解析が行われ，(M)型構文解析では最右導出を前提とした構文解析が行われる．
- (L)型構文解析の例としては(N)構文解析，(M)型構文解析の例としては，(O)構文解析がある．この方法では，アクション表と GOTO 表をもとにして shift と reduce の操作を行いながら構文解析が行われる．このうち reduce は，生成規則の(P)辺に対応する部分を見つけて(Q)辺に置き換えるという(R)操作に対応している．
- 文法とその文法にマッチした文字列に対する計算手続きを与えて，プログラミング言語の構文解析器を作るソフトウエアのことを(S)と呼ぶ．この例としては，Unix や Linux 上の(T)やこれを更に拡張した Bison などがある．これらの(S)で主に用いられているのは(U)という講文解析アルゴリズムである．
- (T)や Bison は単体で用いられることはほとんどなく，字句解析用オートマトンを生成する(V)や flex と併用されて，(C)やインタープリタが作成される．
- 言語は文法によって規定された(W)記号列の(X)集合である．

２．　下記の TAS 命令と等価な if 文を用いない TAS 命令を右の欄に示しなさい．(5 点)

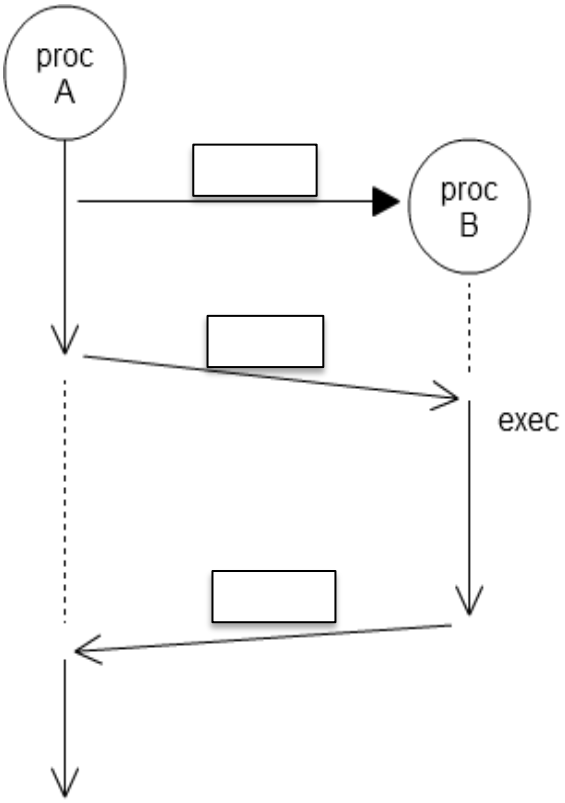
```
int TestAndSet(int* Vaddr){
        if (*Vaddr) return 0;
        else { *Vaddr=0;
                        return 1;
        }
}
```

```
int TestAndSet(int* Vaddr){



}
```

３．　下記は，Unix におけるプロセスの生成過程を表している．このとき，下の問いに答えなさい．(5 点)



・図中の［　　］中に適切なシステムコール名を入れなさい．

・proc A がシェルのプロセスであった場合，このようなプロセスの発生を伴うジョブを，何ジョブと呼ぶか？

［　　］ジョブ

・下記 a-d のうち，正しいものに○をしなさい．

「procB は exec システムコールによって，

a. プロセスが生成される　b.実行が開始される

c. 実行内容が変化する　d. 資源割り当てが起きる.」

3.　以下の正規表現（A～D）に対応する記号の集合（すなわち言語）L(A)～L(D)があるとき，L(A)～L(D)の関係はどのようになっているかを BEN 図で表しなさい．但し，　a,b は終端記号を表わし，答えは L(A)～L(D)の関係を表す BEN 図のみで良い．(10 点)

A:　a*b*
B:　aa*b*
C:　a*bb*
D:　(a|b)*

4.　下記文法のもとで，「3-(1+1)-1」の構文解析を行った結果を，導出木，構文木の形で表わしなさい。（導出木７＋構分木３＝10 点）

<算術式> →<算術式><加減演算子><項>|<項>

<項>→ <項><乗除演算子><因子>|<因子>

<因子>→ <変数>|<数値>|(<算術式>)

<数値>→1|2|… |9

<変数>→A|B|C|…|Z

<加減演算子>→+|－

<乗除演算子>→×|÷

7.　下記の文法から右に示すアイテム集合を得た．アイテム集合の空欄を全て埋め，図を完成させなさいなさい．また，この図のうち，衝突が起きる集合を列挙し，その理由を述べなさい(20 点)

(1) <E> →<E>+<B>

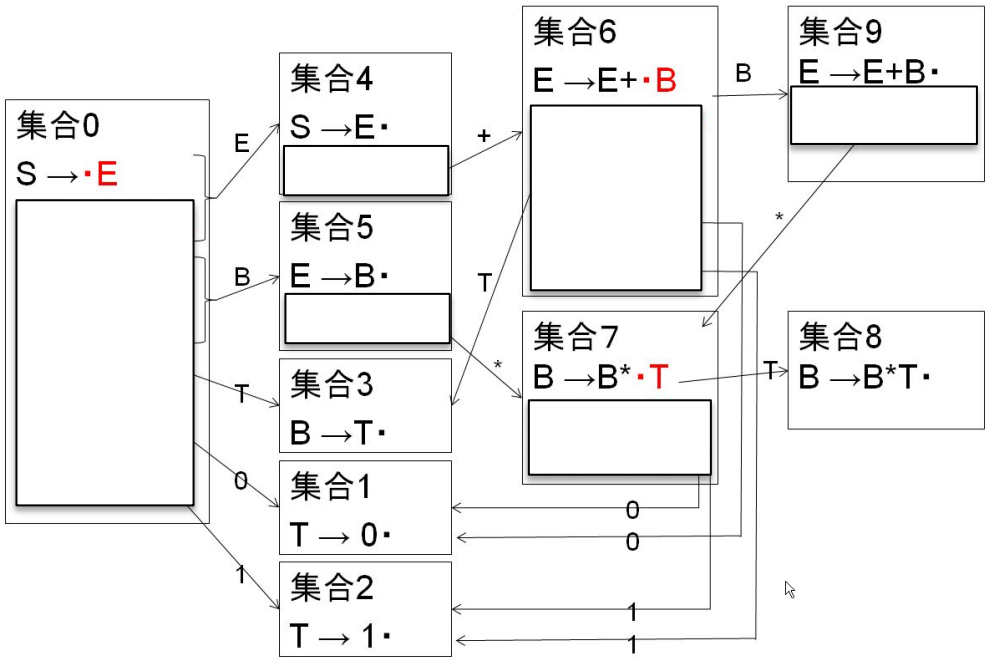
(2) <E> →<B>

(3) <B> →<B>*<T>

(4) <B> →<T>

(5) <T> → 0

(6) <T> → 1



衝突が起きるアイテム集合は

_____

である．

理由：_____

_____

_____

_____

2014 年 2 月 5 日システムソフトウエア試験問題解答例

学科　　学籍番号＿＿＿＿＿＿＿＿＿＿　　氏名＿＿＿＿＿＿＿＿＿＿

１．　つぎの文章の空白を埋めなさい．（各 1 点：合計 50 点）

アーキテクチャ・OS（26 点）

- コンピュータは，CPU(中央処理装置)とメモリ，およびその他の入出力装置から構成されている．このうち，CPU は命令の読み取りとデコードなどを行う(A) 制御部と，実際の計算を行う(B) 演算部の２つから構成されている．(A)部には，演算対象や，読み取った命令と変数などを格納する(C) レジスタと呼ばれる小規模で高速なメモリが配置されている．
- CPU は命令の「(D) 読み取り」と「(E) 実行」を繰り返す．しかし，これでは(A)部と(B)部が交互にしか動かず，効率が良くない．これを解決するために，一つ前の命令の(E)と次の命令の(D)を同時に行うことによって高速化を実現する(F) パイプライン処理が一般的に用いられている．
- (G) フォン・ノイマン型のコンピュータは，メモリ上のプログラムを順次読み取り実行する．このために，次の命令読み取り位置を指しているのが，(H) プログラムカウンタという特殊な(C)である．
- 計算結果に応じて(H)の値を書き換える，つまり，実行するプログラムのメモリ上の位置を変えるのが(I) 条件分岐命令である．この命令を読み取った場合，前述の(F)処理で命令を読み取る場所が確定しないため，特別な機構を持たない CPU では計算速度が低下する．
- 現代的な CPU の中には，制御部・演算部の組であるコアが複数備わった(J) マルチコア，と呼ばれる構造のものがある．これらの CPU では同一時刻に異なる命令が同時に実行できる．このような処理を(K) 並列処理と呼ぶ．これに対して，複数のプログラムを短い時間間隔で順次コアに割り当てて実行する(L) 時分割処理により，各時刻では単一のプログラムしか実行していないが，ある程度の時間粒度で見ると複数のプログラムが同時に実行されているように見せることができる．これを並行処理と呼ぶ．
- 実行中のプログラムは，(M) プロセスと呼ばれ，OS からメモリやファイル，CPU 割り当ての優先度などの(N) 計算資源の割り当てを受ける．(M)はさらに(O) スレッドと呼ばれる単位に分けられて実行される．
- 補助記憶装置の一部を利用して，物理メモリの大きさよりも大きな論理メモリ空間を実現する手法を(P)　仮想記憶と呼ぶ．この実現方式には，(M)が確保している物理メモリ単位で，補助記憶装置に退避・復帰させる(Q)　プロセス・スワップ　と，「(R) ページ」と呼ばれる固定サイズのメモリブロック単位で，補助記憶装置とメモリとの間で退避・復帰をさせる(S)　デマンド・ページング　の２種類がある．
- (S)では，物理メモリの空き容量が減った時点で補助記憶装置に不要なメモリを退避させておくという処理が必要になる．これを行うのが，(T)　ページアウト　・デーモンである．このデーモンの動作を規定するアルゴリズムには，古い(R)の置き換え優先度を高くする(U) FIFO　，最も長い期間使われていない(R)の置き換え優先度を高くする(V)　LRU などのアルゴリズムがある．

- ファイルは，補助記憶装置上のデータブロックをつなげた構造を持つ．このつなげ方は２種類あり，データブロック内に次のデータブロックを指すポインタを書く (W)　リンク構造と，データブロックを指すポインタの一次元並びを用いる(X)　インデックス構造がある．
- HDD の場合，データブロックは HDD 上の(Y) セクタに記録されたデータになる．
- (W)構造のファイルシステムの例として FAT がある．クラスタサイズ 16KB の FAT16 の場合，最大ファイルサイズは(Z) 1 GB になる．

プログラム言語処理系（24 点）

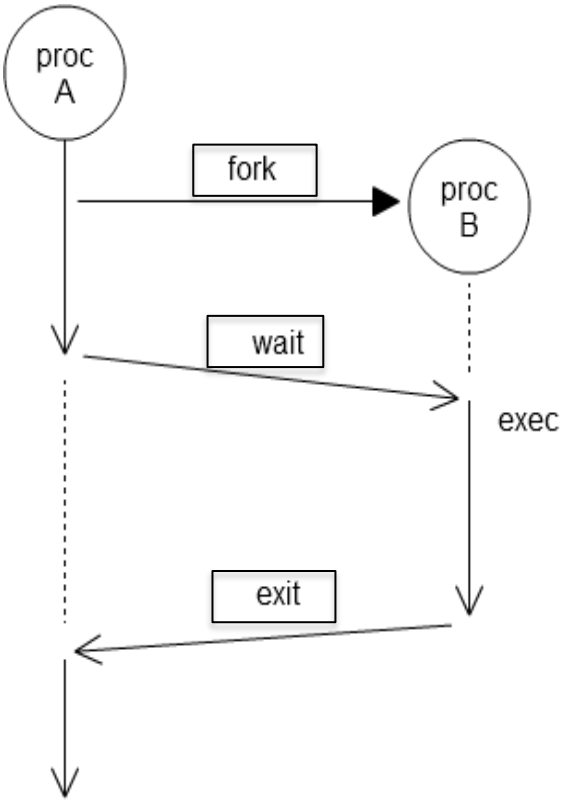
- プログラム言語処理系は，C, C++, Pascal, Basic,などの(A) 高級言語で書かれたプログラムテキストを，(1)CPU が解釈実行可能な(B) 機械語に変換する(C) コンパイラや，(2)逐次解釈・実行するインタープリタ，などで用いられる．
- プログラムテキストは，(D) 正規文法で表現可能な字句の集まりとして表現可能であり，プログラムテキストからこれを抽出する字句解析のためにオートマトンが用いられる．
- 字句解析の後には，構文解析が行われる．通常は(E) 文脈自由文法を用いてプログラミング言語を定義することができる．
- 構文解析では，(F) 出発記号に対してどのような順序で(G) 生成規則が適用され，入力として与えられた終端記号の列が生成されたのかを分析する．この分析結果は(H) 導出木で表すことができる．
- (H)木は，分岐のない中間節をまとめ，演算子を分枝節に引き上げることで(I) 構文木（演算子木とも呼ばれる）に変換することができる．
- 算術式から得られた(I)木を(J) 後順走査することにより(K) 逆ポーランド記法を生成し，これを入力としてスタック上で式の計算を行うことができる．
- 構文解析の方法は大きく分けて(L) 下降型の構文解析と，(M) 上昇型の構文解析法がある．(L)型構文解析では最左導出を前提とした文法の下での構文解析が行われ，(M)型構文解析では最右導出を前提とした構文解析が行われる．
- (L)型構文解析の例としては(N) LL 構文解析，(M)型構文解析の例としては，(O) LR　構文解析がある．この方法では，アクション表と GOTO 表をもとにして shift と reduce の操作を行いながら構文解析が行われる．このうち reduce は，生成規則の(P)右　辺に対応する部分を見つけて(Q) 左　辺に置き換えるという(R) 還元　操作に対応している．
- 文法とその文法にマッチした文字列に対する計算手続きを与えて，プログラミング言語の構文解析器を作るソフトウエアのことを(S)　コンパイラ･コンパイラと呼ぶ．この例としては，Unix や Linux 上の(T)　Yacc やこれを更に拡張した Bison などがある．これらの(S)で主に用いられているのは(U)　LALR という講文解析アルゴリズムである．
- (T)や Bison は単体で用いられることはほとんどなく，字句解析用オートマトンを生成する(V) lex や flex と併用されて，(C)やインタープリタが作成される．
- 言語は文法によって規定された(W) 終端 記号列の(X) 無限集合である．

３．　下記の TAS 命令と等価な if 文を用いない TAS 命令を右の欄に示しなさい．(5 点)

```
int TestAndSet(int* Vaddr){
        if (*Vaddr) return 0;
        else { *Vaddr=0;
                        return 1;
        }
}
```

```
int TestAndSet(int* Vaddr){
        int OldValue;
        OldValue = *Vaddr;
        *Vaddr = 1;
        return !OldValue;
}
```

４．　下記は，Unix におけるプロセスの生成過程を表している．このとき，下の問いに答えなさい．(5 点)



・図中の［　　］中に適切なシステムコール名を入れなさい．

・proc A がシェルのプロセスであった場合，このようなプロセスの発生を伴うジョブを，何ジョブと呼ぶか？

［フォアグラウンド］ジョブ

・下記 a-d のうち，正しいものに○をしなさい．

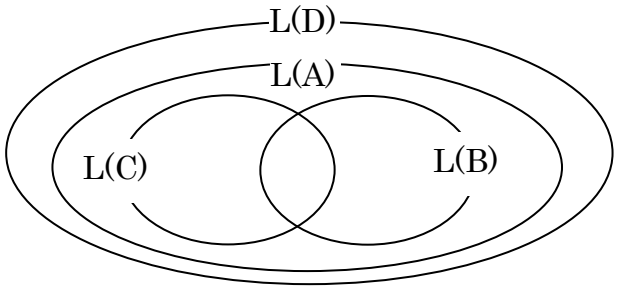
「procB は exec システムコールによって，

a. プロセスが生成される　b.実行が開始される

(c.)実行内容が変化する　d. 資源割り当てが起きる.」

4.　以下の正規表現（A～D）に対応する記号の集合（すなわち言語）L(A)～L(D)があるとき，L(A)～L(D)の関係はどのようになっているかを BEN 図で表しなさい．但し，　a,b は終端記号を表わし，答えは L(A)～L(D)の関係を表す BEN 図のみで良い．(10 点)

A:　a*b*
B:　aa*b*
C:　a*bb*
D:　(a|b)*



5.　下記文法のもとで，「3-(1+1)-1」の構文解析を行った結果を，導出木，構文木の形で表わしなさい。（導出木７＋構分木３＝10 点）

<算術式> →<算術式><加減演算子><項>|<項>

<項>→ <項><乗除演算子><因子>|<因子>

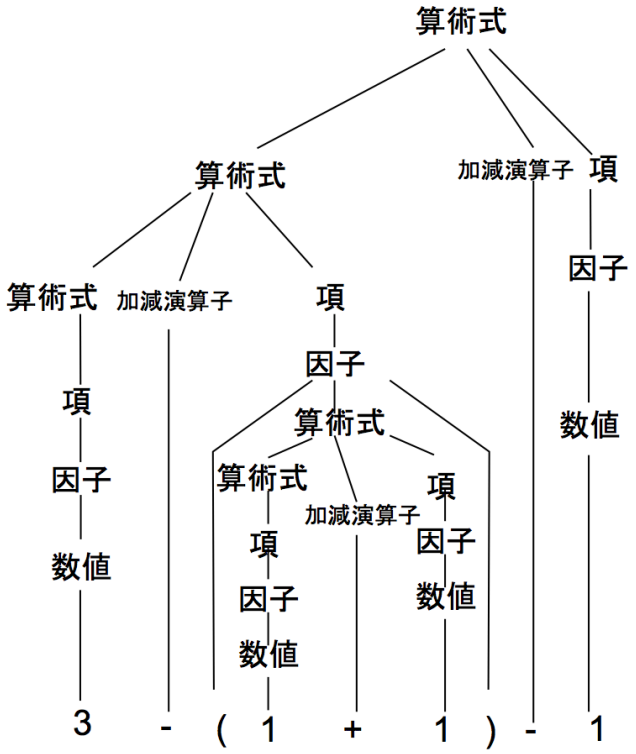
<因子>→ <変数>|<数値>|(<算術式>)

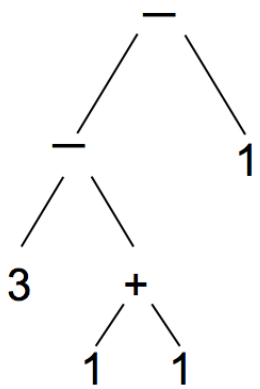
<数値>→1|2|… |9

<変数>→A|B|C|…|Z

<加減演算子>→+|－

<乗除演算子>→×|÷





8.　下記の文法から右に示すアイテム集合を得た．アイテム集合の空欄を全て埋め，図を完成させなさいなさい．また，この図のうち，衝突が起きる集合を列挙し，その理由を述べなさい(20 点)

(1) <E> →<E>+<B>

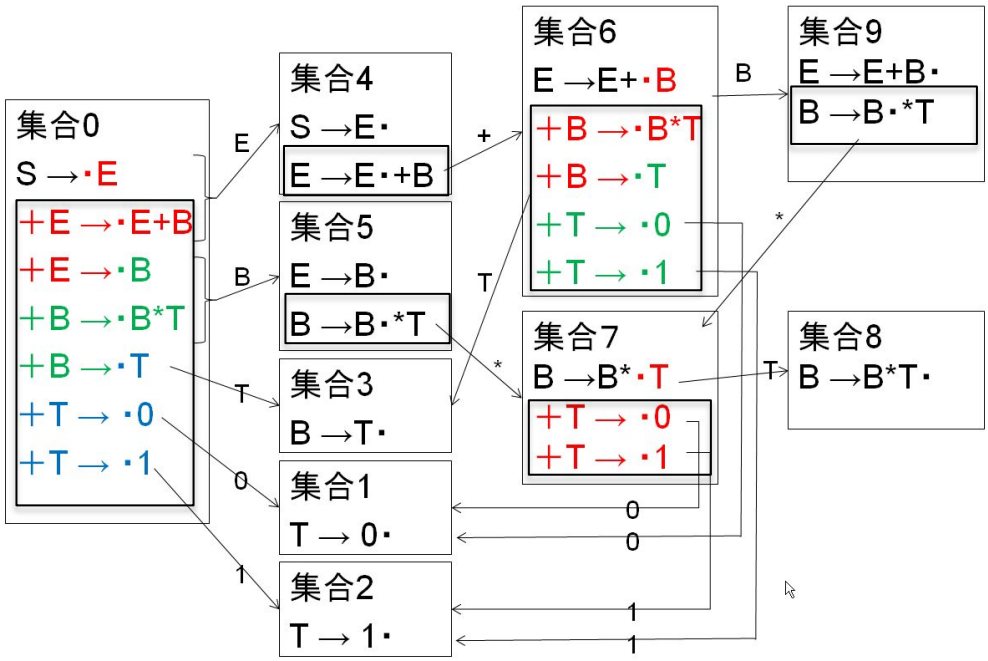
(2) <E> →<B>

(3) <B> →<B>*<T>

(4) <B> →<T>

(5) <T> → 0

(6) <T> → 1



衝突が起きるアイテム集合は

集合５　　と　　集合９

である．

理由：集合５には，“E→B・”と，“B→B・＊T”の要素があり，前者の場合は文法２を用いた還元が必要であり，後者の場合は，“*”を待ち shift7 とすべきである．このように，B を受け取った時点では reduce2 と shift７かが分からないので，shift-reduce conflict が起きる．集合 9 でも同様の理由で reduce1 と shift7 の衝突が起きる．