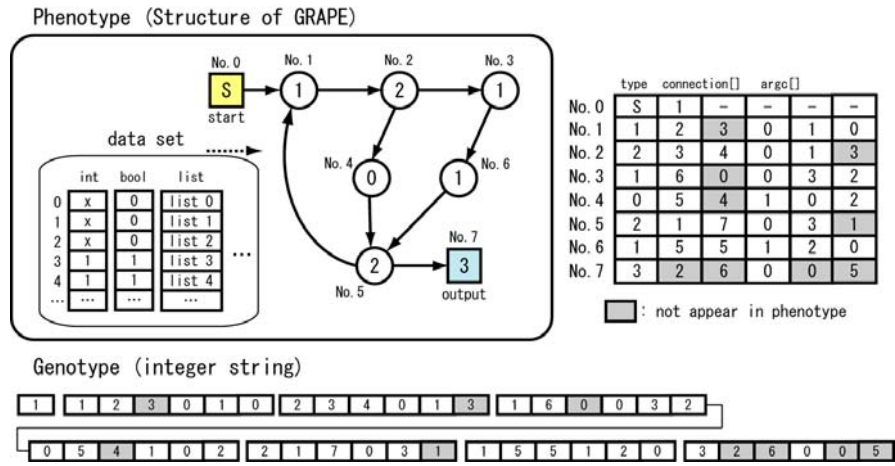


# プログラム自動構築法 GRAPE

- **GRAPE: GR**aph structured Program Evolution は、進化計算に基づき、学習用データセットを与えるだけで**複雑なプログラムを自動構築する方式**である。GRAPEにおける表現型・遺伝子型間マッピングを下図に示す。各ノードはデータに対する演算や分岐の機能をもつ。

- GRAPEの設計指針・特徴を次に示す。

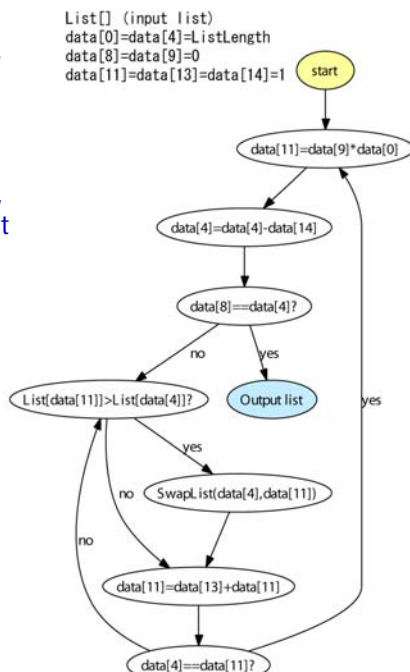
- **モジュール性:**
  - グラフ構造による分岐、ループの表現
- **複数のデータ型**
  - “データセット”の使用による複数データ型の扱い
- **効率の良い自動構築**
  - 整数列の遺伝子に対する遺伝操作



白川真一, 長尾智晴:GRAPE: GRaph structured Program Evolution によるプログラムの自動生成, 電気学会論文誌C, Vol.29, No.3, pp.70-80 (2008)  
 Shinichi Shirakawa, Shintaro Ogino and Tomoharu Nagao: Graph Structured Program Evolution; Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007), Vol.2, pp.1686-1693, London, England, Jul.7-11 (2007)

# プログラム自動構築の実験例

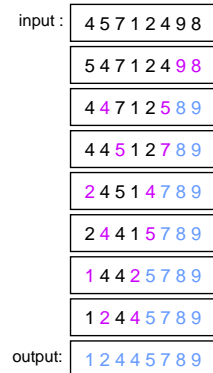
- これまでに、従来のGP(遺伝的プログラミング)では自動生成することができない様々なプログラムを、GRAPEを用いて自動構築できることを確認している。
- ここではその一例としてリストのソートプログラムの自動生成例を示す。実験条件は次の通りである。
  - ノード関数: +, -, \*, <, >, ==, SwapList, EqualList, GreaterList, LessList, Outlist
  - 学習データ: 長さ10~20のリスト30例  
例) {3,4,1,5,9,2,7,10,8,6} → {1,2,3,4,5,6,7,8,9,10}
  - ノード総数: 最大値を与えて自動設定
- 自動構築された構造を右に示す。使用ノード数は8個である。学習データ以外の任意のデータもソート可能である。
- 今後改良を加え、さらに複雑なプログラムの自動構築を目指す。プログラムを人が試行錯誤で作る必要がない時代も夢ではない?!



```

data[4]=ListLength;
data[11]=1;
while(1) {
  data[11] = 0;
  data[4] = data[4] - 1;
  if(data[4] == 0) {
    return List[];
  }
  else {
    do {
      if(List[data[11]] > List[data[4]])
        SwapList(data[4], data[11]);
      data[11] = data[11] + 1;
    } while(data[4] == data[11]);
  }
}
    
```

等価なCプログラム



選択ソート(Selection Sort)に近い  
 具体的な処理の流れ