

## 1.5 Конструирание на самопораждаща се програма (Quine).

Лема за разгъването  $\{ P(Q\_P)=Q\_P() \}$ .  
Нерешимост на stop-задачата.

### 1.5.1 Конструирание на самоизписваща се програма.

#### Задача:

Да се напише програма (на любимият език за програмиране на пишещият), която да изписва на стандартния изход себе си (собственият си текст). Програмата не трябва да чете информация от никакви входни устройства (т.е. започва работа при празна работна памет).

#### Решение:

Любимият ми език е Pascal (по-специално FreePascal за Linux), затова ще се опитвам да реша задачата на този език. Ще кръстя програмата Self.

... досещане ...

Ето и първото ми решение:

```
program self0;
const l=13; ast=chr($27);
var a:array[1..l] of string; i:word;
begin
  a[ 1]:=' a[';
  a[ 2]:=']:=[';
  a[ 3]:=']';
  a[ 4]:='program self0;';
  a[ 5]:='const l=13; ast=chr($27);';
  a[ 6]:='var a:array[1..l] of string; i:word;';
  a[ 7]:='begin';
  a[ 8]:=' for i:=4 to 7 do writeln(a[i]);';
  a[ 9]:=' for i:=1 to l do writeln(';
  a[10]:='   a[1],i:2,a[2],ast,a[i],ast,a[3]);';
  a[11]:=' for i:=8 to l do writeln(a[i]);';
  a[12]:=' { Do something else here }';
  a[13]:='end.';
  for i:=4 to 7 do writeln(a[i]);
  for i:=1 to l do writeln(
    a[1],i:2,a[2],ast,a[i],ast,a[3]);
  for i:=8 to l do writeln(a[i]);
  { Do something else here }
end.
```

### 1.5.2 Особенности на самоизписващата се програма. Варианти.

### 1.5.3 Самопораждане в Pascal. Лема за разгъването. Оператор Quine.

$$P(Q\_P) =_{\text{Pas}} Q\_P()$$

### 1.5.4 Самопораждане в $TM_2$ . Явна конструкция.

### 1.5.5 Самопораждане в $TM_2$ . Използване на компилатор P2TM.

Нека с P2TM именуваме програма на Pascal, която приема като входни данни каква да е програма на Pascal и я компилира до еквивалентна машина от  $TM_2$ . Възможността за написването на подобен компилатор следва от тезиса на Чърч. Поради техническата му сложност ще приемем, че

някой трудолюбив програмист ни го е предоставил наготово.

Нека имаме каква да е машина  $Alg_{TM}$  от  $TM_2$ , очакваща като входни данни описание на машина от  $TM_2$ . Да кръстим с  $Alg_{Pas}$  аналогичната ѝ програма на Pascal (ще е вярно равенството  $P2TM(Alg_{Pas})=Alg_{TM}$ ). Да означим с  $Q_{Pas\_P2TM|Alg_{Pas}}$  диагонала на програмата на Pascal  $P2TM|Alg_{Pas}$ . Да означим програмата  $P2TM(Q_{Pas\_P2TM|Alg_{Pas}})$  с  $Q_{TM\_Alg_{TM}}$ . Имаме равенствата:

- (1a)  $Alg_{Pas}(Q_{TM\_Alg_{TM}}) =_{Pas} Alg_{Pas}(P2TM(Q_{Pas\_P2TM|Alg_{Pas}}))$  определение на  $Q_{TM\_Alg_{TM}}$
- (2a)  $Alg_{Pas}(P2TM(Q_{Pas\_P2TM|Alg_{Pas}})) =_{Pas} (P2TM|Alg_{Pas})(Q_{Pas\_P2TM|Alg_{Pas}})$  определение на композиция "|"
- (3a)  $(P2TM|Alg_{Pas})(Q_{Pas\_P2TM|Alg_{Pas}}) =_{Pas} Q_{Pas\_P2TM|Alg_{Pas}}()$  лема за разгъване

От свързването на горните три равенства следва:

- (1b)  $Alg_{Pas}(Q_{TM\_Alg_{TM}}) =_{Pas} Q_{Pas\_P2TM|Alg_{Pas}}()$
- (2b)  $P2TM(Alg_{Pas})(Q_{TM\_Alg_{TM}}) =_{TM} P2TM(Q_{Pas\_P2TM|Alg_{Pas}})()$  сводимост чрез P2TM
- (3b)  $Alg_{TM}(Q_{TM\_Alg_{TM}}) =_{TM} P2TM(Q_{Pas\_P2TM|Alg_{Pas}})()$  определение на  $Alg_{Pas}$
- (4b)  $Alg_{TM}(Q_{TM\_Alg_{TM}}) =_{TM} Q_{TM\_Alg_{TM}}()$  определение на  $Q_{TM\_Alg_{TM}}$

Последното равенство ни дава търсения Quine оператор за  $TM_2$ . В частния случай, когато  $Alg$  е нищо неправеща програма (идентитета  $I$ ), получаваме  $Q_{TM\_I} = P2TM(Q_{Pas\_P2TM})$ , за която ще е изпълнено:

- (1c)  $I_{TM}(Q_{TM\_I}) = Q_{TM\_I}()$  следствие от (4b)
- (2c)  $Q_{TM\_I} = Q_{TM\_I}()$  свойство на идентитета

От (2c) следва, че програмата  $Q_{TM\_I}$  е самоизписваща се машина на Тюринг.