

نرم افزار تبدیل فلوجارت به برنامه کامپیوتری معادل

سید مهدی شریعت زاده

SMSariatZ@yahoo.com

چکیده مقاله:

فلوجارت به دلیل استفاده از عبارات زبان های طبیعی به عنوان فرمانها و ترسیمی بودن، همچنان کاربردهای آموزشی و علمی خود را حفظ کرده است. در این طرح، الگوریتمی ارائه شده است که با دید تئوری گراف به فلوجارت، عمل تبدیل آن به برنامه کامپیوتری را به طور نرم افزاری (و نه به صورت دستی و توسط کاربر) انجام دهد. در پیاده سازی عملی نیز این تبدیل به دو زبان آموزشی و علمی بیسیک و پاسکال به طور کامل صورت می گیرد.

کلمات کلیدی: فلوجارت-بلوک-تبدیل به برنامه-گراف-جستجوی سطحی گراف-برنامه نویسی بهینه

۱- مقدمه:

جهت تبدیل فلوجارت به برنامه از فرم استاندارد برای یک برنامه ساخت یافته استفاده می شود. شکل کلی این فرم به صورت زیر است:

[دستورهای انضمام فایل های لازم]

عنوان برنامه

تعریف متغیرها

تعاریف زیربرنامه ها

[دستور کنترل خطا]

بخش اصلی برنامه

با توجه به اینکه بلوک های فلوجارت بیانگر مفهوم یک دستور هستند، لذا از این فرم کلی می توان برای تبدیل فلوجارت به هر زبان برنامه نویسی استفاده کرد. البته این ساختار در بعضی از زبان ها دچار تغییر می شود. بعنوان مثال در زبان پاسکال باید کلیه زیربرنامه ها قبل از بخش اصلی (main) قرار گیرند، در صورتی که در بیسیک چنین نیست. همچنین کاراکترهای قابل قبول و عملگرهای زبان های مختلف متفاوت می باشند. در نتیجه این ساختار کلی به فرم های خصوصی تری برای هر زبان تبدیل می شود. در ذیل، ساختار فلوجارت های تبدیل شده به برنامه در حالت کلی برای دو زبان بیسیک و پاسکال به طور دقیق تر نمایش داده شده است. (خطوط تیره نشان دهنده تکرار خط بالا می باشند.)

فلوجارت ابزاری جهت طراحی منطقی برنامه های کامپیوتری است که بیانگر ماهیت اصلی یک برنامه می باشد. امروزه با توجه به گسترش برنامه نویسی و لزوم آموزش آن نقش فلوجارت نیز به عنوان ابزار مولد برنامه هر روز پررنگ تر می شود. این مطلب به علت شکل نموداری و ترسیمی آن و واضح بودن ترتیب مراحل مختلف فلوجارت مشهودتر است. علاوه بر این، می توان سطح (Level) فلوجارت را از کلیه زبان های برنامه نویسی سطح بالا نیز بالاتر دانست، زیرا فرمان های آن بسیار به زبان طبیعی نزدیک بوده و درک عملکرد بلوک های گرافیکی آسانتر از کدهای برنامه است. این دلایل باعث شده که فلوجارت هم به جهت کاربرد آموزشی و هم به عنوان یک زبان سطح بالا برای نمایش دادن الگوریتم ها همچنان کاربرد خود را داشته باشد.

نوشتن برنامه های کامپیوتری را می توان در سه مرحله کلی طبقه بندی کرد: درک مسأله، طراحی الگوریتم و فلوجارت راه حل، و پیاده کردن الگوریتم با استفاده از دستورهای یک زبان برنامه نویسی. هدف از این طرح، حذف کردن مرحله آخر از مراحل فوق است. این مطلب خصوصاً با توجه به کاربرد آموزشی فلوجارت ها می تواند جهت آموزش برنامه نویسی بسیار مفید باشد. در این طرح سعی شده است تا با ارائه یک الگوریتم با استفاده از تئوری گراف این عمل به طور اتوماتیک انجام شود.

۲- ساختار برنامه خروجی:

گره شرط که ۲ و گره پایان که ۰ می باشد) لزوماً ۱ است. این گراف می تواند دارای یک یا چند حلقه نیز باشد.

از دیدگاه برنامه نویسی، بلوک شروع معادل هیچ دستوری نمی باشد. نماد پایان برابر با دستور پایان و بلوک های دیگر نیز مترادف با یک دستور در زبان برنامه نویسی مقصد هستند، اما فلش های ارتباطی فلوجارت که یال های این گراف را تشکیل می دهند به صورت های متفاوتی ظاهر می شوند. این موضوع با توجه به اینکه بلوک های شرط بیش از یک مسیر خروجی دارند پیچیده تر می شود، به طوری که در حالاتی در برنامه ظاهر نمی شوند و یا اشکال مختلفی از جمله دو نوع حلقه مشروط را ایجاد می کنند. به همین جهت پشت سر ها قرار دادن دستور های معادل بلوک ها نمی تواند روش مفیدی برای این منظور باشد، زیرا احتمال تبدیل نادرست و یا ایجاد حلقه بی پایان وجود دارد. همچنین با رسیدن به بلوک شرط، تعیین ادامه مسیر و دستور های دو حالت شرط ناممکن است. در نتیجه تنها بلوکی که سبب ایجاد حالت های مختلف می شود، بلوک شرط است. با توجه به اینکه بلوک های شرط دارای دو انشعاب خروجی هستند، می توانند در قسمت هایی از گراف حلقه ایجاد کنند و یا به صورت شرط معمولی به کار روند. این حلقه ها توسط الگوریتم «جستجوی سطحی» نظریه گراف [۴ و ۶] قابل شناسایی می باشند.

به منظور روشن تر شدن این موضوع، ساختار های فوق در ۵ دسته زیر طبقه بندی شده اند:

۳-۱- شرط ها

بلوک های برنامه شرطی در سه دسته مختلف طبقه بندی می شوند. قابل ذکر است که همگی انواع شرط را می توان در یک دسته قرار داد، ولی جهت بهینه سازی (Optimize) برنامه به وجود آمده این تفکیک صورت گرفته است. در غیر این صورت ممکن است در برنامه ایجاد شده، با بلوک های خالی مواجه شویم که از نظر برنامه نویسی بهینه قابل قبول نیست. [۲] مانند:

```
If x>0 Then
    Print "X is greater than 0"
Else
End If
```

الف) شرط های دارای هر دو دسته فرمان درستی و

نادرستی

ساختار پاسکال	ساختار بیسیک
{عنوان برنامه}	عنوان برنامه
Var	نوع As متغیر Dim
; نوع: متغیر	-----
-----	-----
Label 1,2,...;	On Error {Goto ErHandle Resume next}
نام زیربرنامه Procedure	دستور های برنامه اصلی
Begin	-----
دستور های زیربرنامه	-----
End;	دستور های [LabelName]:
	پرش
Begin {Main}	-----
دستور های برنامه اصلی	دستور کنترل [ErHandle]:
-----	خطا]
-----	نام زیربرنامه Sub
-----	دستور های زیربرنامه
End.	End Sub

توابع اصلی تبدیل فلوجارت به برنامه، ExportToBasic و ExportToPascal می باشند که برنامه تبدیل شده را برمی گردانند و ساختارهایی که نمایش داده شده است در این دو تابع پیاده سازی شده اند.

در قسمت هایی که تبدیل قسمتی از فلوجارت یا تعدادی از بلوک ها به برنامه مورد نیاز است، دو تابع ExportToBasicSomeObjs و ExportToPascalSomeObjs این عمل را انجام می دهند. دو تابع فوق با گرفتن یک محدوده از فلوجارت که توسط دو بلوک آغاز و پایان آن مشخص می شود، برنامه معادل را بر می گردانند که بخش اعظم برنامه لازم جهت تبدیل و کلیه الگوریتم های بخش ۴ در آنها قرار دارد.

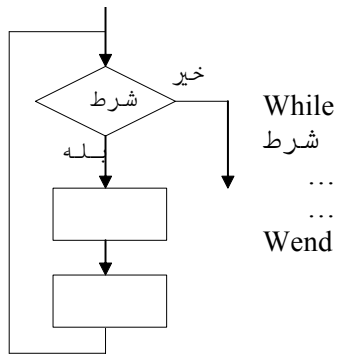
۳-بررسی فلوجارت از دیدگاه تنوری گراف

از نظر تنوری گراف هر زیربرنامه فلوجارت یک گراف جهت دار است که بلوک ها حکم گره های آن را دارند و مسیرهای میان آنها نیز یال های گراف هستند. این گراف جهت دار از یک گره واحد با نام گره «شروع» آغاز می شود و به یک یا چند گره «پایان» ختم می گردد. درجه ورودی هر گره عددی بزرگتر از صفر است (بجز گره شروع که برابر با صفر است) و درجه خروجی آنها (بجز

۲-۳- حلقه ها

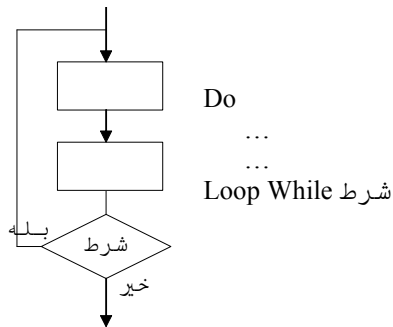
در همه زبان های سطح بالا، دو نوع حلقه شرطی با عملکرد یکسان در نظر گرفته شده است که تفاوت آنها در محل قرار گرفتن شرط می باشد و دو دستور مختلف برای آغاز و پایان این دو نوع حلقه تعریف می شود:

الف) حلقه هایی که شرط آنها بعد از دستورهای بدنه قرار دارد.



الف-2

ب) حلقه هایی که شرط آنها قبل از دستورهای بدنه قرار دارد.



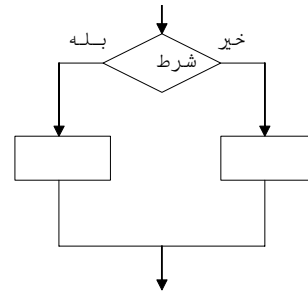
ب-2

While-Wend

Do-Loop While

۴- الگوریتم تبدیل فلوجارت به برنامه

تبدیل فلوجارت به برنامه از پیدا کردن نماد شروع در نمودار گردش آغاز می شود. در ادامه براساس فلش هایی که میان بلوک ها ترسیم شده است بلوک های بعدی (به ترتیب منطقی اجرا) هرکدام به دستور معادل در زبان مقصد تبدیل می شوند. عمل تبدیل



If شرط Then

...

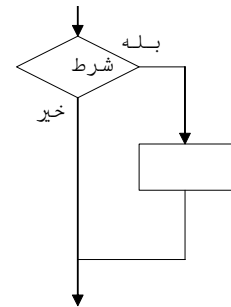
Else

...

End If

الف-۱

ب) شرط های دارای فقط دستورهای درست بودن شرط



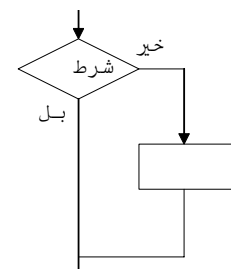
If شرط Then

...

End If

ب-۱

ج) شرط های دارای فقط دستورهای نادرست بودن شرط



If not(شرط) Then

...

End If

ج-۱

شکل ۱) انواع حالات شرط در فلوجارت: الف-شرط کامل

ب-شرط فاقد فرامین عدم صحت شرط ج- شرط فاقد فرامین صحت شرط

بار توسط خودشان فراخوانی می شوند. سپس تورفتگی اول خطوط به اندازه ۲ کاراکتر بیشتر می شود.

در دو تابع فوق برای یافتن بلوک بعد از آخرین بلوک تبدیل شده و همچنین تشخیص حلقه ها، از الگوریتم جستجوی سطحی استفاده شده است. تابعی به نام NextObject با گرفتن شماره هر بلوک، بلوک بعدی (به ترتیب مسیر فلوجارت) را پیدا کرده و شماره آن را بر می گرداند. این کار از طریق یافتن خطی که از بلوک اول آغاز می شود و پیدا کردن سمت دیگر این خط انجام می شود.

هرکدام از زیربرنامه های فلوجارت نیز توسط دو تابع فوق به صورت مشابه به برنامه تبدیل می شوند و با توجه به ساختار کلی برنامه های تبدیل شده (بخش ۲)، در محل مناسب خود قرار می گیرند.

۵- نتیجه گیری:

این الگوریتم با استفاده از زبان Visual Basic 6 در ۸ سابروتین و ۹۱۸ خط پیاده سازی شده است. روش فوق در اجرای عملی قادر است با استفاده از اطلاعات فلوجارت های ترسیم شده توسط همین برنامه، تبدیل به برنامه کامپیوتری معادل به دو زبان بیسیک و پاسکال را انجام دهد. لازم به ذکر است برنامه کامل ترسیم، اجرا و تبدیل فلوجارت که دریافت و طبقه بندی اطلاعات بلوک ها و فلش ها را نیز انجام می دهد از ۱۹ پنجره و ۳۷۵۶ خط برنامه ایجاد شده که برای بررسی عملکرد آن به تحلیل یک فلوجارت نمونه می پردازیم.

جهت نمایش خروجی این روش، برنامه های کامپیوتری معادل فلوجارت محاسبه انتگرال (شکل ۳) به دو زبان بیسیک و پاسکال که با استفاده از این روش ایجاد شده است، در لیست ۱ و ۲ مشاهده می شود:

هر بلوک به جمله معادل (بجز شرط ها) برای این زبان توسط دو تابع ExportToBasicObj و ExportToPascalObj انجام می شود که با گرفتن شماره هر بلوک، متن دستور معادل آن را برمی گردانند. این عمل با توجه اینکه اطلاعات زیادی از هر بلوک در هنگام طراحی فلوجارت، در فایل ذخیره می شود بسیار ساده است. سپس کلیه بلوک های توضیح جستجو می شوند تا در صورت لزوم توضیحات درج شده بعد از خط فوق به شکل مناسب (برای برنامه های پاسکال، درون { } و برای زبان بیسیک، بعد از ') در فایل نوشته شوند. الگوریتم تشخیص جملات شرطی و نوع آن به صورت زیر است:

* - با توجه با اینکه تنها بلوک های شرط داری دو انشعاب خروجی هستند، پس هر بلوک از این نوع موجب پیدایش یکی از پنج حالت ذکر شده می گردد و با رسیدن به چنین بلوکی، اقدامات زیر صورت می گیرد:

۱- از یکی از مسیرهای خروجی شرط، جستجوی سطحی آغاز می شود و همه بلوک های بعدی فلوجارت شماره گذاری می شوند.

۲- از انشعاب دوم خروجی شرط، جستجوی دیگری انجام می شود و تا هنگامی که به یک بلوک که در مرحله قبل شماره گذاری شده است برسد، ادامه پیدا می کند. شماره این بلوک ذخیره می شود. این بلوک، اولین دستور بعد از خاتمه شرط یا حلقه است.

۳- اگر در حین انجام جستجوی دوم به دستور شرط اولیه رسیدیم، این شرط مولد یک حلقه است.

در صورتی که به دستورهایی که قبل از شرط تبدیل کرده بودیم برسیم، شرط در پایان حلقه و نوع الف است و در غیر این صورت نوع ب می باشد.

۴- اگر پس از انجام شماره گذاری در بند ۲، بلوک شرط شماره ای دریافت نکند، حلقه ای به وجود نیامده است.

با احضار دوباره تابع متن مربوط به قسمت برقراری شرط و بخش خلاف بودن شرط را به دست می آوریم. در صورتی که هیچ کدام از دو رشته فوق خالی نبودند، شرط از نوع الف است و اگر یکی از این دو خالی باشد، حالات شرط ب یا ج اتفاق می افتد و براساس هرکدام از حالت های فوق دستور مناسب انتخاب شده و به پایان برنامه نوشته شده اضافه می گردد.

۵- در صورت وقوع هرکدام از حالات فوق، برای تبدیل دستورهای درون حلقه و یا فرامین شرط ها که هرکدام جمعاً یک بلوک برنامه (بلوک درون حلقه یا دو بلوک صحت یا خلاف شرط) را تشکیل می دهند، توابع فوق یک

لیست ۱: برنامه پاسکال معادل فلوجارت انتگرال

{Program Integral - Exported from
FlowChart in ۱۰/۱۰/۲۰۰۳}

Var

Deghat : Real;

Dmin : Real;

Dmax : Real;

h : Real;

x : Real;

FI : Real;

D1 : Integer;

D2 : Integer;

Begin {Main}

f:=""

Write('Enter function: f(x)=');

Readln(f);

Write('Enter range[D1,D2]:');

Readln(D1,D2);

Write('Enter Deghat:');

Readln(Deghat);

x:=D1;

FI:=eval(f);

h:=(D2-D1)/Deghat;

Repeat

x:=x+h;

FI:=FI+eval(f);

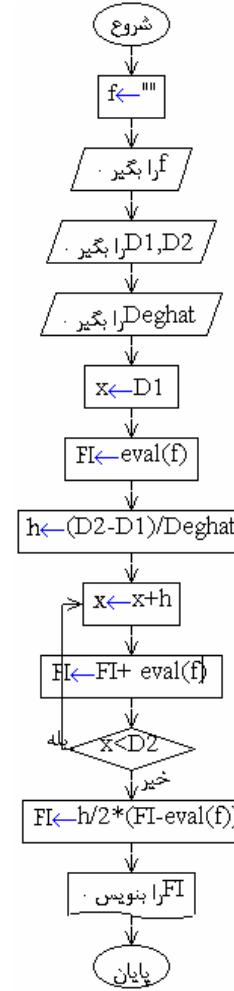
Until not(x<D2);

FI:=h/2*(FI+eval(f));

Writeln('Integral of f(x) in [D1,D2]:',FI);

Readln;

End.



شکل ۳: فلوجارت محاسبه انتگرال

[1] MSDN Library CDs

[۲] آشنایی با Turbo Pascal 7 ، تألیف مهندس محمد سلیمی و

کیوان عاملی، مجتمع فنی تهران، چاپ هشتم، تهران ۱۳۷۸

[۳] طراحی و تحلیل الگوریتم ها، تألیف مونس علی فرمنش، جهاد

دانشگاهی واحد صنعتی امیرکبیر، تهران ۱۳۷۸

[۴] آشنایی با تئوری گراف و کاربردهای آن، سید مهدی شریعت

زاده، ارائه شده در فراخوان مقاله ریاضی استان خوزستان(حائز رتبه

اول)، بهار ۱۳۷۹

[۵] الگوریتم و فلوجارت، تألیف بهرام غلامی و علیرضا جباریه،

چاپ دوم، نشر مجتمع فنی تهران، تهران ۱۳۷۷

[۶] ریاضیات گسسته و ترکیباتی، تألیف رالف. پ. گریمالدی جلد دوم

و سوم، انتشارات فاطمی، تهران ۱۳۷۸

'Program Integral - Exported from FlowChart in

28/06/2001

Dim Deghat As Integer

Dim h As Double

Dim x As Double

Dim FI As Double

Dim D1 As Integer

Dim D2 As Double

Cls

Input "Enter function: f(x)=" ,f

Input "Enter range[D1,D2]:" ,D1,D2

Input "Enter Deghat:" ,Deghat

x=D1

FI=eval(f)

h=(D2-D1)/Deghat

Do

 x=x+h

 FI=FI+eval(f)

Loop While x<D2

FI=h/2*(FI-eval(f)*2)

Print "Integral of f(x) in [D1,D2]=" ,FI

End