

Alphi

Naam:: Lissens
Voornaam:: Tobiah
Richting:: 2de bachelor Informatica
year :: 2016-2017

Inhoud

1. Inleiding
2. Syntax (BNF)
3. Semantiek
4. VoorbeeldProgramma's
5. Demo _police
6. Demo _line
7. Demo _ultra
8. Implementatie
9. Data Definitie
10. Parsen
11. Evalueren
12. Robot Lib
13. Conclusie
14. Algemeen
15. Definitie
16. Implementatie
17. Index
18. AlphiExamples
19. Src

Inleiding

In dit project wordt de eenvoudige programmeertaal Alphi opgesteld. Hierbij is het de bedoeling verschillende basiselementen van een imperatieve programmeertalen te implementeren. zoals bv: toekenning, variabelen en volgorde van bewerkingen. De taal die hieronder wordt uitgewerkt heet Alphi wat staat voor alphanumerical. Deze taal maakt enkel gebruik van alphanumerische karakters met de uitzondering dat whitespace ook is toegestaan. Eerst zal de syntax worden vastgelegd. Vervolgens wordt de semantiek vastgelegd en worden voorbeeld programma's gegeven. Hierna worden de implementatie aspecten besproken.

Syntax

`Pgm = Stmt`

```

Stmt    = <Var> "Is" <Exp> "Stop"
        | "Command" <Output> <Exp> "Stop"
        <Stmt> <Stmt>
        "If" <Exp> "Begin" Stmt "End"
        "While" <Exp> "Begin" Stmt

Exp      ::= <BExp>
        | <NExp>
        | "Command" <Input>

NExp     = <Num>
        <NVar>
        <NExp> "Add" <NExp>
        <NExp> "Sub" <NExp>
        | <NExp> "Mul" <NExp>
        | <NExp> "Div" <NExp>
        | "Open" NExp "Close"

BExp     = <Bool>
        | <BVar>
        "Not" <BExp>
        <NExp> "Gt" <NExp>
        | <NExp> "Lt" <NExp>
        | <NExp> "Eq" <NExp>
        <BExp> "And" <BExp>
        | <BExp> "Or" <BExp>
        | "Open" BExp "Close"

Input    ::= "OpenMBot"
        | "CloseMBot"
        | "SensorR"
        | "SensorL"
        | "Ultra"

Output   ::= "Print"
        | "MotorR"
        | "MotorL"
        | "Led1"
        | "Led2"

Bool     = "True" | "False"
Num      = Int | Float
Int      ::= ["0"-"9"]
Float    ::= <Int>"Point"<Int>
Var      = NVar | BVar

```

```
NVar    ::= "N"<Letter>+
BVar    ::= "B"<Letter>+
Letter  ::= ["a"-"Z"]
```

Semantiek

1. Expressies

2. Numeric

- Volgorde van bewerkingen

Voor bewerkingen op hetzelfde niveau wordt van links naar rechts geëvalueerd.

Hoe hoger het niveau hoe eerder ze geëvalueerd worden

Niveau 1(literalen)

Float, Int

Vb: 10Point4, 4

Niveau 2

Add, Sub

Niveau 3

Mul, Div, Mod

Niveau 4 (haakjes)

Open Close

vb: 4 Sub 4 Mul 4 = -12

4 Mul 4 Mul 2 Sub 3 = 29

3. Boolean

Volgorde Van bewerkingen

Niveau 1

True, False

Niveau 2

And, Or

Niveau 3

Gt, Lt , Eq

Niveau 4

Not

Niveau 5 (Haakjes)

```

vb: True And True           = True
    4 Lt 5 And False        = False
    4 Lt 5 And True         = True
    4 Lt 5 And False Or True = True

```

4. Statements

5. AssignStatement ““ Usage: Ken een waarde aan een variable Toe

Zet variable Tobiah op 20 vb:

2. While

Usage: Herhaalt de statements tussen Begin en End tot de Expression voor Begin naar False evalueert

```

Vb: Programma dat tot 10 telt.
Nx IS 0
While Nx Lt 10 Begin
    NX Is NX Add 1 Stop
End

```

3. If

Usage: Voert Stuk code tussen Begin en End uit indien de Expressie voor Begin naar True evalueert.

vb: Programma dat bij oneven getallen 1 optelt. If Nvar Mod 2 Eq 1 Begin
Nvar Is Nvar Add 1 Stop End

3. Commands

Programma's

Korte beschrijvingen van het programma

1. demo_police.alp

Start teller.

Indien teller even zet Led1 op rood en led 2 op blauw.

Indien teller oneven zet led2 op rood en led1 op blauw.

Verhoog Teller met 1

Begin bij stap 2.

2. demo_line.alp

Lees beide lichtsensoren uit.

Indien beide sensoren Zwart zien rij de robot rechtdoor.

Indien links wit ziet en rechts zwart draai alleen de linker motor.

Indien rechts wit ziet en links zwart draai alleen de rechter motor.
Indien Beide wit zien rij achteruit.
Begin terug bij stap 1.

3. demo_ultra.alp

Lees Ultrasonesensor uit.

Indien afstand Groter dan 40 rij rechtdoor.

Indien afstand Kleiner dan 40 draai de linkermotor vooruit en de rechtermotor achteruit.

Begin terug bij stap 1.

Implementatie

1. Parsen

2. Evalueren

Conclusie

1. Algemeen:

Een alphanumerical taal maken leek in het begin leuk. Dit bracht echter enkele nadelen met zich mee. Het groote nadeel hieraan is dat je geen special karakters hebt die kunnen instaan voor bv het ei. Verder wordt de taal ook Enorm rap onduidelijk en on leesbaar doordat er weinig tot geen ondersch

2. Syntax definitie:

Hierbij zijn er soms onlogische samenstellingen mogelijk zoals Numerical Expression in de IfConc. Hierdoor moet dit opgevangen worden tijdens het evalueren dit is ongewild.

3. Implementatie:

De parseLibrary is vrij onduidelijk geschreven er ontbreekt een mooie volgbare hierachy die bv we. Dit komt voornamelijk doordat er geen eenduidige manier was om dingen te parsen En er op ieder mo

Appendix Broncode:

Inhoud

1. AlphiExamples

1. demo_police.alp

2. demo_line.alp

3. demo_ultra.alp

2. Src

1.

2.

3.

4.
5.
6.
7.
8.
9.
10.
11.

AlphiExamples

1. demo_police.alp

```
0 commentOpen 1 2 A simple police sirene program 3 4 commentClose 5 6 7
Command OpenMBot Stop 8 9 Ncount Is 0 Stop 10 While True Begin 11 12
If Ncount Mod 2 Eq 0 Begin 13 Command Led1 1 Stop 14 Command Led2 3
Stop 15 End 16 17 If Ncount Mod 2 Eq 1 Begin 18 Command Led1 3 Stop 19
Command Led2 1 Stop 20 End 21 22 Ncount Is Ncount Add 1 Stop 23 24 End
25 Command CloseMBot Stop
```

2. demo_line.alp

```
0 commentOpen 1 2 A simple linefollowing program 3 4 commentClose 5 6 7
Command OpenMBot Stop 8 While True Begin 9 10 Bleft Is Command SensorL
Stop 11 Bright Is Command SensorR Stop 12 13 If Bleft And Bright Begin 14
Command MotorL 70 Stop 15 Command MotorR 70 Stop 16 End 17 18 If Bleft
And Not Open Bright Close Begin 19 Command MotorL 0 Stop 20 Command
MotorR 80 Stop 21 End 22 23 If Bright And Not Open Bleft Close Begin 24
Command MotorL 80 Stop 25 Command MotorR 0 Stop 26 End 27 28 If Not
Open Bright Or Bleft Close Begin 29 Command MotorL 0 Sub 60 Stop 30
Command MotorR 0 Sub 60 Stop 31 End 32 End 33 Command CloseMBot
Stop
```

3. demo_ultra.alp

```
0 commentOpen 1 2 A simple wall evade program 3 4 commentClose 5 6 7
Command OpenMBot Stop 8 9 While True Begin 10 Ndistance Is Command
Ultra Stop 11 12 If Ndistance Gt 40 Begin 13 Command MotorL 70 Stop 14
Command MotorR 70 Stop 15 End 16 17 If Ndistance Lt 39 Begin 18 Command
MotorL 70 Stop 19 Command MotorR 0 Sub 70 Stop 20 End 21 Command Print
Ndistance Stop 22 End ““
```

Src