Universitas Esa Unggul

Smart, Creative and Entrepreneurial

www.esaunggul.ac.id

CMC 101 TOPIK DALAM PEMROGRAMAN

PERTEMUAN 3

PROGRAM STUDI MAGISTER ILMU KOMPUTER

FAKULTAS ILMU KOMPUTER

TOPIK DALAM PEMROGRAMAN
PEMROGRAMAN FUNGSIONAL

Pertemuan 3

TUJUAN PERKULIAHAN

- Mahasiswa mampu membuat berpikir abstrak dan fungsional, dan mampu membuat program sederhana dalam bahasa pemrograman Haskell

- Abstraksi dan “dekomposisi” dalam konteks fungsional: data (type bentukan), fungsi
- Ekspresi aritmatika, logika, dan kondisional
- Analisis rekurens
- Konsep list sebagai struktur rekursif
- Operasi dasar list dengan elemen tertentu: integer, character, type bentukan

Lisp

- Lisp, being the second oldest programming language in the world (after Fortran), still holds a top position in AI creating due to its unique features. For example, Lisp has a special macro system which makes possible to develop a domain specific level of abstraction and build the next level on it. Lisp in artificial intelligence development is known for its unique flexibility as it adapts to the problem you need to solve on the contrary to the other languages that are chosen because they can complete this or that task. Developers opt for Lisp in machine learning and inductive logic projects.
- *Features:*
  – fast prototyping capabilities;
  -support for symbolic expressions;
  – automatic garbage collection which actually was invented for the Lisp language;
  – library of connection types including dynamically-sized lists and hastables;
  – efficient coding due to compilers;
  – interactive evaluation of components and recompilation of files while the program is running.

Haskell

- Haskell is a purely functional programming language that can boast about its lazy evaluation and type interface features. LogicT monads facilitate expressing non-deterministic algorithms, and algorithms can be expressed in a compositional way.
- *Features:*
  – major algorithms available via cabal;
  – CUDA binding;
  – compiled to bytecode;
  – can be executed on multple CPU in cloud.

AIML

- AIML (Artificial Intelligence Markup Language) is a dialect of XML used to create chatbots. Due to AIML one can create conversation partners speaking a natural language.
  The language has categories showing a unit of knowledge; patterns of possible utterance addressed to a chatbot, and templates of possible answers. To know how it works check out this article about building a chatbot.
- So, the matter of best-something is rather philosophical in any sphere, and AI development is not an exception. There are a lot of factors influencing the choice of programming languages for an AI project. It depends on functions you need to create, usage and even your taste in some cases. However, more and more AI programmers are using Python as it's a simple and powerful tool, while C++, Prolog and Lisp can be called runners-up in this race.

Pemrograman Fungsional

Bahasa pemrograman fungsional : Lisp, Scheme, ML, Haskell

PROGRAMMING FUNGSIONAL

Bahasa pemrograman fungsional:

- Disebut aplikatif karena fungsi yang di aplikasikan ke dalam argumentasi menjadi deklaratif dan non prosedural

- Didasarkan pada konsep matematika dari sebuah fungsi dan bahasa pemrograman fungsional, meliputi :
  - suatu set fungsi primitif
  - suatu set format fungsional
  - aplikasi operasi
  - suatu set objek data dan fungsi asosiasi
  - suatu mekanisme untuk memberikan rujukan sebuah nama terhadap suatu fungsi

- Merupakan hasil dari fungsi meringkas dan men-generalisir type data dari peta

3 komponen primer bahasa functional :

- **Kumpulan objek data**
  menggunakan mekanisme struktur data tingkat tinggi.
  Contoh : Array atau List

- **Kumpulan fungsi built-in**
  untuk memanipulasi objek data dasar yang menyediakan sejumlah fungsi untuk membuat dan mengakses list.

Contoh :

LISP → - bahasa untuk komputasi simbolik, nilai direpresentasikan dengan ekspresi simbolik.

- banyak digunakan di wilayah kecerdasan buatan (robotika, sistem cerdas)
- biasa dieksekusi dibawah kendali interpreter

Ekspresi terdiri dari atom atau list.

Atom → string dan karakter (huruf, angka)

Contoh : A 68000

List → urutan dari atom atau list, dipisahkan dengan spasi, ditutp dengan tanda kurung

Contoh : (PLUS A B)

((DAGING AYAM) (SAWI KANGKUNG BAYAM) AIR))

ML (Meta Language) →

- merupakan bahasa aplikatif dengan program-program yang ditulis menggunakan gaya C atau PASCAL dan dengan konsep yang lebih advance tentang tipe data
- mendukung polymorphisme dan abstraksi data
- berjalan dengan interpreter

- **Kumpulan functional forms**

  untuk membuat fungsi baru, yang mengizinkan programmer mendefinisikan operasi baru dari kombinasi fungsi yg ada

LAMBDA CALCULUS

Adalah :

- bahasa sederhana dengan ilmu semantik sederhana, ekspresif yang menyatakan semua fungsi dapat diperhitungkan
- Merupakan suatu bentuk formal dengan fungsi sebagai aturan

Contoh :

- Dengan ekspresi polynomial X^2+3X-5
- Dengan fungsi lebih dari 1 variabel

  (+ x y) ditulis ((+ x) y) dimana fungsi (+ x) adalah fungsi yang menambahkan sesuatu ke x

Lambda Calculus murni mempunyai 3 buah Elemen :

- Lambang primitif
- Aplikasi fungsi
- Fungsi ciptaan

Lambda calculus murni tidak mempunyai fungsi tetap atau konstanta

Kalkulasi dalam lambda calculus adalah :

menulis ulang (mengurangi) suatu lambda-expression menjadi suatu format formal

ILMU SEMANTIK OPERASIONAL

Inti denotasional Ilmu Semantik adalah :

Terjemahan dari program konvensional ke dalam persamaan fungsional.

Tujuan denotasional Semantik dari suatu bahasa adalah :

menugaskan suatu nilai kepada setiap ekspresi dalam bahasa

Ilmu Semantik dapat dinyatakan dalam lambda calculus sebagai fungsi matematical, Eval, dari ekspresi ke nilai

Contoh : Eval[+ 3 4]=7 menggambarkan bhw nilai ekspresi (+ 3 4) untuk menjadi 7

FUNGSI REKURSIF

Perluasan Syntax Lambda-calculus yang mencakup ekspresi yang telah dinamai (named expressions)

L ::= ...| x : L | ...

X = nama dari Ekspresi Lambda L

FAC : \n.(if (= n 0) 1 (* n (FAC (- n 1))))

dengan **syntactic sugaring** :

FAC : \n.if (n = 0) then 1 else (n * FAC (n - 1))

FAC : (\fac.(\n.(if (= n 0) (* n (fac (- n 1))))) FAC)

H : \fac.(\n.(if (= n 0) 1 (* n (fac (- n 1)))))

FAC : (H FAC)

ATURAN LINGKUP LEKSIKAL

let n : E in B
adalah penyingkatan untuk (\n.B) E

let x : 3 in (* x x)

\y. let x : 3 in (* y x) Ekuivalen \y. (* y 3)

letrec n : E in B
adalah penyingkatan untuk let n : Y (\n.E) in B

let n : E in B = (\n.B) E
letrec n : E in B = let n : Y (\n.E) in B

SEMANTIK TRANSLASI DAN KOMBINATOR

Kombinator :

S = \f .(\g .(\x. f x (g x)))

K = \x .\y. x

I = \x.x

Y = \f. \x.(f(x x)) \x.(f (x x))

aturan reduksi untuk kalkulus SKI adalah :

S f g x --> f x (g x)

K c x --> c

I x --> x

Y e --> e (Y e)

(A B) --> A B

(A B C)--> A B C

- Aturan Reduksi dijalankan dari kanan ke kiri
- Jika tidak ada reduksi S,K,I,Y maka tanda kurung akan dibuang, dan proses reduksi diteruskan

Semantik Translasi untuk Lambda Calculus :

Compile [s] --> s

Compile [(E1 E2)] --> (Compile [E1] Compile [E2])

Compile [\x.E] --> Abstract [(x, Compile [E])]

Abstract [(x, s)] --> if (s=x) then I else (K s)

Abstract [(x, (E1 E2))] --> ((S Abstract [(x, E1)]) Abstract [(x, E2)])

Dimana s adalah simbol

Scheme

- Turunan dari LISP, didasarkan pada Lambda Calculus. Dikonsentrasikan ke fitur lambda-calculus

- Scheme mempunyai dua object :
  - Atoms : Untaian Karakter yang bukan blank
  - List : Rangkain Atom atau List dipisahkan oleh blank dan berada dalam tanda

- Sebuah fungsi dapat terbuat atas fungsi yang lain dan dapat diaplikasikan pada list atau argumen

DARI SISI SEJARAH

ALTERNATIF TEORI DASAR MATEMATIKA

- **Alonso Church, lambda-calculus,1930an**
- **Haskell B. Curry,logika kombinatorial**

1958, LISP (LISt Processing), pemrosesan list berdasarkan fungsi rekursif.

- Recursion
- First class function
- Garbage collection

1960an, penggunaan lambda-calculus di dalam ilmu komputer → Semantik Denotasional

Teori Semantik Formal untuk bahasa pemrograman

(Peter Landin, Christopher Strachy, dll)

1969, Model Matematika pertama untuk lambda-calculus bertipe bebas (Dana Scott)

Haskell

- **bahasa modern yang dinamai sama dengan Haskell B. Curry**
- **Didesain oleh 15 orang anggota komite internasional**
- **Pembentukan bahasa fungsional yang memasukkan :**
  - **ide-ide baik yang sebelumnya ada dalam riset bahasa fungsional**
  - **Yang sesuai untuk pengajaran, riset dan aplikasi**
  - **Fasilitas Overloading, yang dipadukan dengan sistem bertipe polimorfis, i/o fungsional, abstraksi data dan penyembunyian informasi**

Functional Programming urutan mesin virtual.

- **Bahasa pemrograman fungsional → lambda calculus**
- **Lambda calculus → logika kombinatorial**
- **Logika kombinatorial → kode mesin reduksi graf**

Kesemuanya adalah mesin virtual

SOAL-SOAL

1. Bahasa untuk komputasi simbolik, nilai yang direpresentasikan dengan ekspresi simbolik disebut :

   a. Atom b. LISP * c. List d. ML

2. Urutan dari atom atau list, yang dipisahkan dengan spasi, dan ditutup dengan tanda kurung adalah :

   a. Atom b. LISP c. List * d. ML

3. Yang tidak termasuk elemen lambda calculus adalah :

   a. lambda primitif b. fungsi ciptaan

   c. ilmu semantik * d. aplikasi fungsi

4. Tujuan denotasional semantik dari suatu bahasa adalah :

a. menulis ulang suatu lambda-ekspression menjadi suatu format formal

b. menugaskan suatu nilai kepada setiap ekspresi dalam bahasa *

c. memanipulasi objek data dasar yang menyediakan sejumlah fungsi untuk membuat dan mengakses list

d. membuat fungsi baru yang mengizinkan programmer mendefinisikan operasi baru dari kombinasi fungsi yang ada

5. Yang termasuk dalam kumpulan objek data adalah :

a. LISP dan ML
b. Array atau List *
c. Atom atau List
d. ML dan Array

6.Prinsip yang terdapat dalam Lambda-Calculus :

a. Nilai suatu expresi tergantung hanya pada nilai dari sub-ekspresinya *
b. Kurang memadai untuk mengekspresikan fungsi komputasi
c. Nilai suatu ekspresi tidak tergantung atas nilai sub-ekspresi
d. Semantik yang kompleks

7.Scheme memiliki 2 buah objek, yaitu :

a. Atom dan List *
b. List dan Semantic
c. Atom dan Semantic
d. Function dan list

8.Manakah yang bukan merupakan bahasa pemrograman fungsional modern?

a. Haskell *
b. LISP
c. Scheme
d. Tidak ada yang benar

9.Sebuah program fungsional terdiri atas

a. Ekspresi E
b. Algoritma
c. Input
d. Tidak ada yang benar

10. Ekspresi Lambda-Calculus $\x.(x^3 - 27)\ 5$ akan menghasilkan :

a. 68
b. 78
c. 88
d. 98