LINGUAGEM CHURCH: MODELO GENERATIVO QUE PRETENDE UNIFICAR AS TEORIAS DA INTELIGÊNCIA ARTIFICIAL

https://doi.org/10.5281/zenodo.15571854

ROMANO, Simone Maria Viana, Especialista*

*Faculdade de Tecnologia de Praia Grande
CEETEPS - Centro Estadual de Educação Tecnológica Paula Souza
Pça. 19 de Janeiro, 144, Boqueirão, Praia Grande / SP, CEP: 11700-100
Fone: (13)3591-6968
simone@fatecpg.com.br

RESUMO

A inteligência artificial, na atualidade vem sendo pesquisada com base em regras probabilísticas ao invés das regras do pensamento, como era realizada nas décadas de 1950 e 1960. Para criar a modelagem probabilística utilizando as linguagens formais, usam-se as técnicas de inferência que ajudam na modularidade e reutilização. Noah D. Goodman (cientista do Departamento de Ciências Cognitivas e Cerebrais do Instituto de Tecnologia de Massachusetts – MIT) criou uma linguagem de programação probabilística chamada Church que descreve os modelos geradores. A semântica da linguagem *Church* foi definida nos termos de histórias de avaliação e distribuições condicionais sobre estas histórias. Este artigo tem o objetivo de mostrar as principais características e o funcionamento dessa nova linguagem de programação que promete revolucionar a inteligência artificial através da teoria unificada.

PALAVRAS-CHAVE: linguagem church, regra probabilística, regra de inferência, modelo probabilístico.

ABSTRACT

Artificial intelligence, today has been researched based on probabilistic rules instead of rules of thought, as it was done in the 1950s and 1960s.To create probabilistic modeling using formal languages, they

use the techniques of inference that help modularity and reuse. Noah D. Goodman (Scientist, Department of Brain and Cognitive Sciences at the Massachusetts Institute of Technology - MIT) has created a programming language called Church probabilistic models describing the generators. The language semantics Church was defined in terms of stories and assessing conditional distributions on these stories. This article aims to show the main features and operation of this new programming language that promises to revolutionize the artificial intelligence through the unified theory.

KEY-WORDS: *language church, probabilistic rule. rule of inference, probabilistic model*

INTRODUÇÃO

No início o pensamento era visto pela inteligência artificial como uma inferência lógica. Por isto, as primeiras linguagens de programação eram baseadas em linguagem matemática e utilizavam frases afirmativas, como por exemplo: "As aves põem ovos, logo, as galinhas são aves". Essas linguagens conseguiam mostrar conceitos distintos, porém, nem todos, como por exemplo, "o ornitorrinco, que não é uma ave, põe ovos".

Nos últimos tempos, as pesquisas vêm avançando na inteligência artificial, com o uso da inteligência probabilística, no qual os computadores podem aprender com a análise dos dados de treinamento (grandes conjuntos de dados) utilizando os padrões estatísticos.

Com essas pesquisas, surgiram, por exemplo, sistemas que realizam a tradução automatizada de textos.

Assim, surge a linguagem Church, desenvolvida pela equipe do cientista Noah D. Goodman, no Instituto Tecnológico de Massachusetts (MIT), uma linguagem de programação que utiliza o raciocínio probabilístico nas regras de inferência.

1 REGRAS DE INFERÊNCIA

Segundo Desessards (2010), inferência é o processo pelo qual se chega a uma proposição, firmada na base de uma ou várias proposições aceitas como ponto de partida do processo. Ou ainda, é a derivação de novas sentenças a partir de sentenças antigas (RUSSELL e NORVIG, 2004).

Uma regra de inferência desejável é que haja um procedimento efetivo para determinar se uma fórmula é ou não inferível. Utiliza-se das seguintes especificações:

a) premissas são regras sem hipóteses dentro de uma inferência dada. Possui duas partes: premissa e conclusão;
b) se a hipótese for verdadeira, então, a conclusão também é;
c) notação simples para comandos de implementação;
d) verificação de tipos é baseada em inferência;
e) o início acontece com um sistema simples de regras e vão sendo adicionados novas características gradualmente.

As principais regras de inferência são: principio de adição; princípio de simplificação; princípio de desligamento; princípio da conjunção; principio da contra-proposição; principio da inferência equivalente; princípio do silogismo hipotético; silogismo disjuntivo; absorção; dilema destrutivo e dilema construtivo.

2 MODELO PROBABILÍSTICO

Um modelo tem a função de descrever um processo no qual os dados são gerados.

Segundo Russel e Norvig (2004), o modelo probabilístico consiste em um espaço amostral de resultados possíveis mutuamente exclusivos, juntamente com a medida de probabilidade para cada resultado. Por exemplo, no modelo de previsão de tempo, os resultados poderiam ser ensolarado, nublado, chuvoso e com neve, formando assim, um subconjunto.

Esse modelo descreve documentos com base nos pesos

binários que mostram a ausência ou presença de termos. O cálculo de probabilidade é resultado gerado pelo modelo. O Teorema de Bayes é a ferramenta matemática principal dessa libguagem e permite que probabilidades desconhecidas sejam calculadas a partir de probabilidades condicionais conhecidas, em geral no sentido causal (VAN RIJSBERGEN, 1979).

O modelo é baseado em *Probability Ranking Principle (*Principio Probabilístico da Ordenação) e informa que a hipótese de relevância de um documento para determinada consulta é independente dos outros documentos. Neste, é considerado o processo interativo de estimativas da probabilidade de relevância e promete um bom desempenho a partir do princípio probabilístico da ordenação. É usado para representar uma estrutura simplificada de um domínio e também responder a perguntas diferentes pla inferência condicional.

Os modelos probabilísticos são muito úteis na ciência cognitiva, inteligência artificial e aprendizado de máquina. Em geral, são implementados por meio das combinações da linguagem natural com a linguagem matemática e a inferência para cada modelo novo. Diferenciando assim da abordagem processual onde o conhecimento é o mapeamento da entrada-saída de uma questão específica.

Segundo Hardesty (2010), Goodman combinou os antigos sistemas baseados em regras com *insights* dos sistemas probabilísticos mais recentes, criando uma nova forma de modelar o pensamento e trazendo assim, grandes impactos na ciência cognitiva e na inteligência artificial. Por isso, surge a linguagem formal *Church* que descreve os modelos generativos e promete unificar as teorias da Inteligência Artificial.

3 LINGUAGEM CHURCH

Church é uma linguagem de programação construída pelo cientista Noah D. Goodman e foi batizada com esse nome, em homenagem a Alonzo Church.

Alonso Church foi um matemático norte-americano que atuou na lógica matemática, teoria da recursão e teoria da computação. Uma das suas contribuições mais famosas é o cálculo lambda, que é um

sistema matemático formal que investiga as funções e aplicação de funções, influenciando as linguagens funcionais, como o LISP, conforme a expressão abaixo que mostra a forma geral da expressão Lamba.[1]

(lambda (x) (+ xx)) → #<procedure>
(lambda (variável) expressao1, expressao2...)

A linguagem Church foi feita a partir da linguagem de programação SCHEME que possui semântica probabilística, cuja criação se deu em 1975, por Guy L. Steele e Gerald Jay Syssman.[2]

A SCHEME é uma linguagem de programação procedural e funcional baseada na linguagem LISP, utilizando o cálculo de lambda (sub-conjunto determinística) para representar modelos de geradores, conforme figura 1 (DYBVIG, 2007):

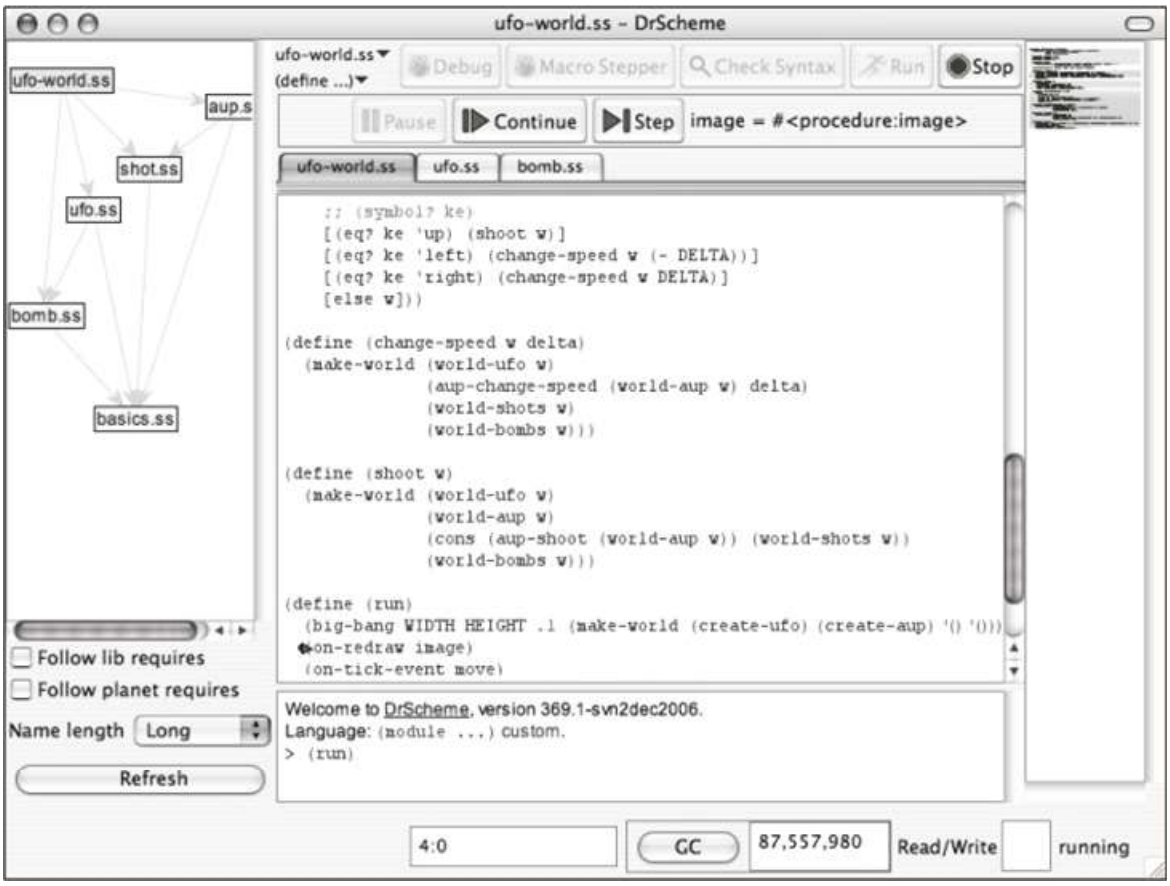


Figura 1 – Exemplo de Programa Scheme

[1] Disponível em: http://www.din.uem.br/ia/precursores/calc_lamb.html Acesso em: 01/05/ 2010.

[2] Language Scheme. Disponível em: http://www.scheme.com. Acesso em: 02/05/2010.

A linguagem Church é uma linguagem universal que descreve os processos generativos e consultas através de condições. Utiliza linguagens formais para a modelagem probabilística e geração de processos, ou seja, é baseada em regras probabilísticas.

Segundo Goodman, a linguagem Church possui as seguintes características:

a) como linguagem LISP, a avaliação e consulta é por amostragem condicional;
b) permite avaliações separadas para compartilhar a história geradora e também permite descrição fácil;
c) sistemas genéricos para inferência exata que implementa uma consulta primitiva, de modo que qualquer programa Church possa ser executado sem escrever o código de inferência.

Por exemplo, temos uma afirmação que um pintassilgo é uma ave. Então um programa desenvolvido em *Church* conclui que pintassilgos podem voar. Porém, caso seja informado ao programa que o peso do pintassilgo é aproximadamente 100 quilogramas, o programa poderá rever a estimativa de probabilidade inicial, concluindo assim que não poderia voar.

Segundo Goodman, um programa em Church que nunca conheceu um pássaro que não pode voar vai definir a probabilidade de que qualquer pássaro voe em 99,99%. Por meio do raciocínio probabilístico, tem-se a estruturação sem muito esforço. Porém, conforme ele vai aprendendo mais sobre o pintassilgo, o programa revisa as probabilidades e encontra novas informações relevantes.[3]

Os pioneiros da inteligência social codificavam manualmente todos os conceitos distintos. Com a linguagem *Church,* o programa aprende essas distinções por si mesmo durante um período (semelhante ao que acontece com o ser humano que vai revendo conceitos antigos baseados em novas experiências).

Segundo Nick Chater, professor de Ciências Cognitivas da Universidade College London, diz que: "... o sistema permite que se

[3] Disponível em: http://www.censanet.com.br/noticias/ler/id-1027577/Cientista_ propoe_teoria_unificada_da_ inteligencia_artificial. Acesso em: 20/ 04/ 2010.

construa um modelo cognitivo de uma forma fantasticamente simples e transparente do que era possível até agora." E complementando, também disse: "... você pode tentar pensar em todas as coisas que um ser humano conhece e tentar listá-las seria uma tarefa sem fim. (...) O sistema *Church* promete que pode – baseado em algumas informações – através de um cálculo probabilístico, traça todas as inferências e consequências."[4]

Há vários modelos probabilísticos classificados como processos geradores. Em resumo, a linguagem *Church* é uma linguagem formal que descreve estes processos. É possível fazer um download através do endereço eletrônico https://svn.csail.mit.edu/mit-church/tags/0.1.2, mas é necessário instalar o SVN e utilizar como usuário e senha a palavra *anonymous*.

O C*hurch* está embutido dentro do *Scheme* e para ser iniciado é necessário incluir a biblioteca *Church* e em seguida chamar o interpretador da linguagem através do macro *church* com o seguinte comando:

```
(import (church))
        (church
          ...your church program...)
```
[5]

Também pode-se executar por meio da interface *on-line*, chamada ChurchServ, *(disponível em http://projects.csail.mt.edu/church/serv/).* No entanto, não há todos os recursos e não deve ser usado em trabalhos longos ou modelagem de informações complexas (ver figura 2).

[4] Disponível em: http://www.censanet.com.br/noticias/ler/id-1027577/Cientista_propoe_teoria_unificada_da_ inteligencia_artificial. Acesso em: 20/ 04/ 2010

[5] Disponível em: http://projects.csail.mit.edu/church/wiki/Church. Acesso em: 30/04/ 2010.

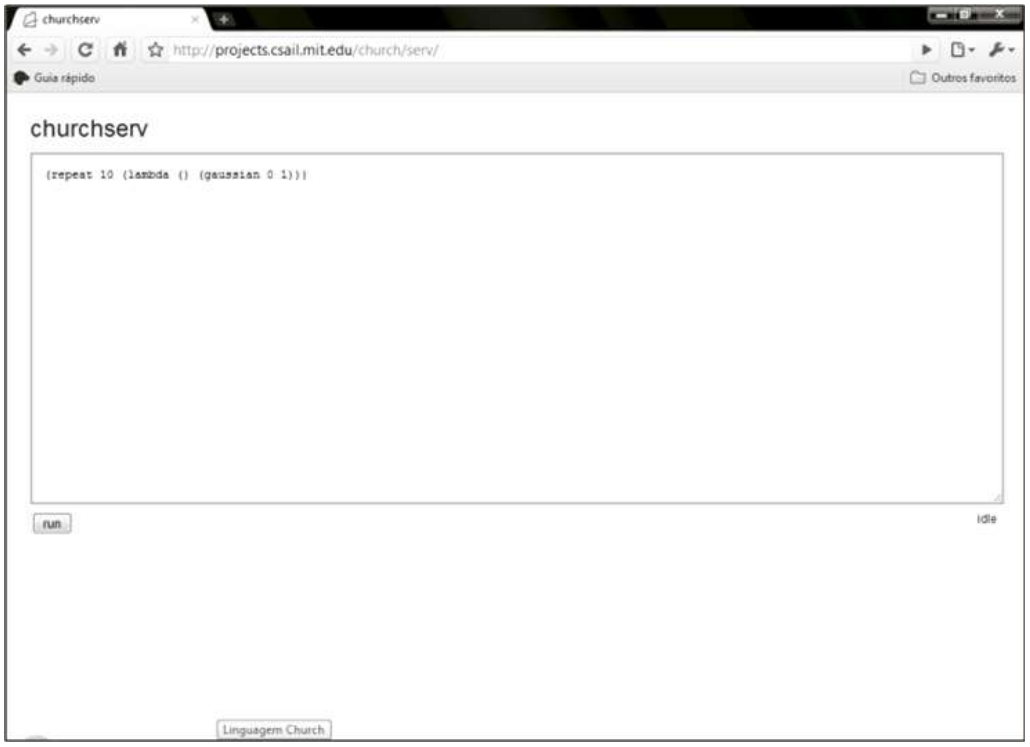


Figura 2 – Tela de Apresentação do ChurchServ

4 CONCLUSÃO

Atualmente, a linguagem Church encontra-se em testes preliminares. A versão *alpha* é instável, lenta, difícil de utilizar corretamente e não terá compatibilidade com as versões anteriores.

Há muito a ser feito e explorado nessa linguagem. Os pesquisadores estão solicitando apoio e aceitam inscrições de grupos de pesquisa de outros países interessados em auxiliar no aperfeiçoamento dessa linguagem que promete revolucionar a inteligência artificial, unificando as regras de inferência, as ciências cognitivas e os modelos probabilísticos.

REFERÊNCIAS BIBLIOGRÁFICAS

CHURCH, Alonso. *Disponível em: <http://www.projects.csail.mit.edu/church/wiki/ Church>. Acesso em: 30/04/ 2010.*

_____ *Disponível em: <http://www.din.uem.br/ia/precursores/calc_lamb.html>. Acesso em: 01/05/ 2010.*

DESESSARDS, J. A. **Lógica, regras e inferência.** *Disponível em: ia.ucpel.tche. br/~lpalazzo/Aulas/IWS/m06/IntroLogica.pdf. Acessado em 30/04/ 2010.*

DYBVIG, R. K. *The Scheme Programming Languague. E ed.* The Mit Press, 2007.

EMACS GNU. *Disponível em: <http://www.gnu.org/software/emacs/>. Acesso em: 03/05/ 2010.*

GOODMAN, D. N.; MASIGNGHKA, V. K. ***Church:*** *a language for generative models*, 2008.

HARDESKTY, L. **Cientista propõe teoria unificada da inteligência artificial.** ***Disponível*** *em: <http://www.censanet.com.br/noticias/ler/id-1027577/Cientista_propoe_teoria_unificada _da_inteligencia_artificial>. Acesso em: 20/04/2010.*

LANGUAGE SCHEME. *Disponível em: <http://www.scheme.com>. Acesso em: 02/05/2010.*

PYTHON PROGRAMMING LANGUAGE. OFICIAL WEBSITE. *Disponível em: <http://www.python.org>. Acesso em: 03/05/2010.*

RUSSELL, S. J. Norvig P. **Inteligência Artificial**. 2 ed. Rio de Janeiro: Campus, 2004.

THE R PROJECT FOR STATISTICAL COMPUTING. *Disponível em: <http://www.r-project.org>/. Acesso em: 03/05/2010.*

VAN RIJSBERGEN, C. J. ***Information Retrieval***. 2nd ed. Butterworths, 1979.