Tabelas de decisão
Pré-processador em Python

Proposta de trabalho

Este trabalho de conclusão de curso propõe o desenvolviemento de um pré-processador para tabelas de decisão (TDs) em Python, em que a abordagem e método de tradução de TDs se baseia na dissertação de Satoshi Nagayama [Nagayama, 1990]. A dissertação apresenta diversos modelos de tradução de TDs, como o `switch method`, que foi utilizado no pré-processador deste trabalho.
Além disso, ressalta-se a relevância da documentação, utilizando uma técnica de auto-documentação do Prof. Dr. Valdemar W. Setzer [Setzer, 1988].

Introdução

A programação, em sua essência, envolve a criação de algoritmos que fazem escolhas com base em condições lógicas. Tradicionalmente, essas escolhas lógicas são implementadas por meio de árvores de decisão com aninhamentos de 'if...then...else', que podem se tornar complexas e difíceis de manter. Nesse contexto, as TDs emergem como uma alternativa poderosa, oferecendo uma forma compacta, gráfica e organizada de especificar escolhas lógicas [Setzer, 2024].

Exemplo de TD

Hierarquia	Gestor	Gestor	Gestor	Gestor	Analista	Analista	Analista	Analista
Setor	Comercial	Comercial	Administrativo	Administrativo	Comercial	Comercial	Administrativo	Administrativo
Metas atingidas	Y	N	Y	N	Y	N	Y	N
Inscrição em cursos de desenvolvimento	1	1	1	1	1	1	1	1
Bonificação de 5% do salário	-	-	-	-	2	2	-	-
Bonificação de 10% do salário	2	2	-	-	-	-	2	2
Bonificação de 15% do salário	-	-	2	2	-	-	-	-
3 dias de folga	3	-	3	-	3	-	3	-

TD no código-fonte

exemplos > tds_traduzidas > poster_TD.py > ...

```
1   hierarquia = ''
2   setor = ''
3   meta_atingida = ''
4
5   def inscreve_em_cursos():
6       print(f'Funcionário inscrito em cursos de desenvolvimento')
7
8   def define_bonificacao(valor:int):
9       print(f'Funcionário recebe bonificacção de {valor}% do salário')
10
11  def atribui_3dias_de_folga():
12      print(f'Funcionário recebe 3 dias de folga')
13
14  if __name__ == '__main__':
15      hierarquia = input('').upper()
16      setor = input('').upper()
17      meta_atingida = (input('').upper() == 'Y')
18      #TD decision table td_estrutura_de_incentivos
19      #TD sets
20      #TD        hierarquia_gestor                  {"GESTOR"}
21      #TD        hierarquia_analista                {"ANALISTA"}
22      #TD        setor_comercial                    {"COMERCIAL"}
23      #TD        setor_administrativo               {"ADMINISTRATIVO"}
24      #TD conditions
25      #TD        hierarquia              hierarquia_gestor   hierarquia_gestor   hierarquia_gestor       hierarquia_ge
26      #TD        setor                   setor_comercial     setor_comercial     setor_administrativo    setor_admini
27      #TD        meta_atingida           Y                   N                   Y                       N
28      #TD actions
29      #TD        inscreve_em_cursos()    1                   1                   1                       1
30      #TD        define_bonificacao(5)   0                   0                   0                       0
31      #TD        define_bonificacao(10)  2                   2                   0                       0
32      #TD        define_bonificacao(15)  0                   0                   2                       2
33      #TD        atribui_3dias_de_folga() 3                  0                   3                       0
34      #TD end table
```

Código gerado

exemplos > tds_traduzidas > poster_TD_processado.py > (ø) hierarquia

```
35      #2 #TD end table
36      #2]
37      def decision_table_td_estrutura_de_incentivos() ->None:
38          I_0 = 0 #Inicialização do auxiliar da condição hierarquia
39          I_1 = 0 #Inicialização do auxiliar da condição setor
40          I_2 = 0 #Inicialização do auxiliar da condição meta_atingida
41          I   = 0 #Inicialização do número da regra
42          if hierarquia in set(["GESTOR"]):
43              I_0 = 0
44          elif hierarquia in set(["ANALISTA"]):
45              I_0 = 1
46          if setor in set(["COMERCIAL"]):
47              I_1 = 0
48          elif setor in set(["ADMINISTRATIVO"]):
49              I_1 = 1
50          if meta_atingida == True:
51              I_2 = 0
52          elif meta_atingida == False:
53              I_2 = 1
54          I = (2*2*1)*I_0 + (2*1)*I_1 + (1)*I_2
55          match I:
56              case 0:
57                  inscreve_em_cursos()
58                  define_bonificacao(10)
59                  atribui_3dias_de_folga()
60              case 1:
61                  inscreve_em_cursos()
62                  define_bonificacao(10)
63              case 2:
64                  inscreve_em_cursos()
65                  define_bonificacao(15)
66                  atribui_3dias_de_folga()
67              case 3:
68                  inscreve_em_cursos()
69                  define_bonificacao(15)
70              case 4:
71                  inscreve_em_cursos()
72                  define_bonificacao(5)
73                  atribui_3dias_de_folga()
74              case 5:
75                  inscreve_em_cursos()
76                  define_bonificacao(5)
77              case 6:
78                  inscreve_em_cursos()
79                  define_bonificacao(10)
80                  atribui_3dias_de_folga()
81              case 7:
82                  inscreve_em_cursos()
83                  define_bonificacao(10)
84              case _:
85                  exit()
86  decision_table_td_estrutura_de_incentivos()
```

Resultados

O pré-processador desenvolvido foi capaz de traduzir TDs em código Python executável, utilizando o método de tradução *switch method*, foi projetado usando o *Strategy Pattern*[Freeman et al., 2004], possibilitando a fácil extensão de novos métodos de tradução; A modularidade do sistema permitiu uma fácil manutenção e evolução do projeto; E a integração de práticas de auto-documentação foi outro marco importante do trabalho por acarretar em uma documentação clara, atualizada e concisa.

Bibliografia

- Freeman, Eric et al. (Nov. 2004). *Head First Design Patterns: A Brain-Friendly Guide*. Sebastopol, CA, USA: O'Reilly Media.
- Nagayama, Satoshi (1990). "Implementação de Gerador I-M-E com Tabelas de Decisão". PhD thesis. São Carlos, SP, Brasil: Instituto de Ciências Matemáticas e de Computação, USP.
- Setzer, Valdemar W. (setembro 1988). *Um Sistema Simples para Documentação Semi-Automática de Programas*. Tech. rep. RT-MAC-8808. Relatório Técnico. São Carlos, SP, Brasil: Instituto de Matemática e Computação, Universidade de São Paulo (USP).
- — (2024). *Comunicação pessoal*. Orientador do Trabalho de Conclusão de Curso.

Aluno: Eduardo Ribeiro Silva de Oliveira
Orientador: Prof Dr Valdemar W. Setzer
eduardo.rso784@usp.br
setzerv@gmail.com
Departamento de Ciência da Computação — Universidade de São Paulo

IME