

Report



Language Translator

Submitted by

Purushottam Kakade

11803394

Programe and Section: B.Tech (CSE), KM072

Course Code: INT248

1- Introduction:-

In this project, a language will be chosen from a list of options in which the text is to be entered, and also the language in which the text is to be translated is also selected from the list of options. After selecting the languages, the translate button will be clicked to translate the text.

Language translators allow computer programmers to write sets of instructions in specific programming languages. These instructions are converted by the language translator into machine code. The computer system then reads these machine code instructions and executes them. Hence, a language translator is a program that translates from one computer language to another.

It was mentioned earlier in this chapter that CPUs could only understand machine code or machine language . Machine code is hardware-specific, and hence there are as many machine codes as there are hardware designs. Although machine code makes complete sense for computers, it is a very difficult and tedious language in which to write programs. Hence programmers developed other less difficult languages in which to write programs. Over the years, these programming languages are moving closer and closer to human language.

2-Code:-

```
from tkinter import *

from translate import Translator

Screen = Tk()

Screen.title("Language Translator with GUI by- TechVidvan")


InputLanguageChoice = StringVar()

TranslateLanguageChoice = StringVar()

LanguageChoices = {'afrikaans': 'af', 'albanian': 'sq', 'amharic': 'am', 'arabic': 'ar', 'armenian': 'hy',
'azerbaijani': 'az', 'basque': 'eu', 'belarusian': 'be', 'bengali': 'bn', 'bosnian': 'bs', 'bulgarian': 'bg', 'catalan': 'ca',
'cebuano': 'ceb', 'chichewa': 'ny', 'chinese (simplified)': 'zh-cn', 'chinese (traditional)': 'zh-tw', 'corsican':
'co', 'croatian': 'hr', 'czech': 'cs', 'danish': 'da', 'dutch': 'nl', 'english': 'en', 'esperanto': 'eo', 'estonian': 'et',
'filipino': 'tl', 'finnish': 'fi', 'french': 'fr', 'frisian': 'fy', 'galician': 'gl', 'georgian': 'ka', 'german': 'de', 'greek': 'el',
'gujarati': 'gu', 'haitian creole': 'ht', 'hausa': 'ha', 'hawaiian': 'haw', 'hebrew': 'he', 'hindi': 'hi', 'hmong': 'hmn',
'hungarian': 'hu', 'icelandic': 'is', 'igbo': 'ig', 'indonesian': 'id', 'irish': 'ga', 'italian': 'it', 'japanese': 'ja',
'javanese': 'jw', 'kannada': 'kn', 'kazakh': 'kk', 'khmer': 'km', 'korean': 'ko', 'kurkish (kurmanji)': 'ku',
'kyrgyz': 'ky', 'lao': 'lo', 'latin': 'la', 'latvian': 'lv', 'lithuanian': 'lt', 'luxembourgish': 'lb', 'macedonian': 'mk',
'malagasy': 'mg', 'malay': 'ms', 'malayalam': 'ml', 'maltese': 'mt', 'maori': 'mi', 'marathi': 'mr', 'mongolian':
'mn', 'myanmar (burmese)': 'my', 'nepali': 'ne', 'norwegian': 'no', 'odia': 'or', 'pashto': 'ps', 'persian': 'fa',
'polish': 'pl', 'portuguese': 'pt', 'punjabi': 'pa', 'romanian': 'ro', 'russian': 'ru', 'samoan': 'sm', 'scots gaelic': 'gd',
'serbian': 'sr', 'sesotho': 'st', 'shona': 'sn', 'sindhi': 'sd', 'sinhala': 'si', 'slovak': 'sk', 'slovenian': 'sl', 'somali':
'so', 'spanish': 'es', 'sundanese': 'su', 'swahili': 'sw', 'swedish': 'sv', 'tajik': 'tg', 'tamil': 'ta', 'telugu': 'te', 'thai':
'th', 'turkish': 'tr', 'ukrainian': 'uk', 'urdu': 'ur', 'uyghur': 'ug', 'uzbek': 'uz', 'vietnamese': 'vi', 'welsh': 'cy',
'xhosa': 'xh', 'yiddish': 'yi', 'yoruba': 'yo', 'zulu': 'zu'}

InputLanguageChoice.set('English')

TranslateLanguageChoice.set('Hindi')

def Translate():

    translator = Translator(from_lang=
InputLanguageChoice.get(),to_lang=TranslateLanguageChoice.get())

    Translation = translator.translate(TextVar.get())

    OutputVar.set(Translation)
```

```

InputLanguageChoiceMenu = OptionMenu(Screen,InputLanguageChoice,*LanguageChoices)
Label(Screen,text="Choose a Language").grid(row=0,column=1)
InputLanguageChoiceMenu.grid(row=1,column=1)

#choice in which the language is to be translated
NewLanguageChoiceMenu = OptionMenu(Screen,TranslateLanguageChoice,*LanguageChoices)
Label(Screen,text="Translated Language").grid(row=0,column=2)
NewLanguageChoiceMenu.grid(row=1,column=2)
Label(Screen,text="Enter Text").grid(row=2,column =0)
TextVar = StringVar()
TextBox = Entry(Screen,textvariable=TextVar).grid(row=2,column = 1)

Label(Screen,text="Output Text").grid(row=2,column =2)
OutputVar = StringVar()
TextBox = Entry(Screen,textvariable=OutputVar).grid(row=2,column = 3)
B = Button(Screen,text="Translate",command=Translate, relief =
GROOVE).grid(row=3,column=1,columnspan = 3)

mainloop()

```

3-Proposed Architecture:-

A compiler or source-to-source translator are tools that read in a text file containing a program and write out another text file containing the translation of that program. For compilers, that generated program is in a low-level language such as assembly language, byte-codes for a virtual machine such as the Java virtual machine (JVM), or sometimes even C. For source-to-source translators the generated program is a higher-level language, which may or may not be the same language as the input program. Thus, the name “compiler” is a bit of a misnomer; it is simply a translator that generates low-level code. Compilers and source-to-source translators are essentially the same kinds of tools built using the same techniques.

Throughout this document we will use the term “translator” to indicate any tool that translates a program from one language, the source language, to another, the target language. In some cases the source and target languages are the same; for example a tool that reformats a program to comply with strict formatting guidelines.

It is common practice to construct language translators as a series or pipeline of components that perform the major tasks in program translation.

Scanning:-

The scanning process reads in program text and recognizes its lexical syntax. This process involve recognizing lexical constructs such as keywords (e.g. while or if), identifier names (e.g. x, area, etc.), literal values (e.g. 1, 3.14, "Hello", etc.) and generating a sequence of tokens, one for each such recognized construct. This sequence of tokens is passed on to the parsing phase that follows. The scanner is also responsible for recognizing and removing comments from the input stream; for these no token is generated.

For example, from the input stream `while (x < 100) x = x * x ;` the following list of tokens may be generated:

```
[ Whilet("while"), LParent("("), Idt("x"), LTt("<"), IntLitt("100"), RParent(")"),  
Idt("x"), EQt("="), Idt("x"), Start("*"), Idt("x"), Semit(";")]
```

Parsing:-

The parsing process recognizes the syntactic structure in the sequence of tokens produced by the scanner. This structure is represented as a tree. For example, the parser would create a tree with a root node labeled to indicate that it was a while-loop with two children; the first being the tree representing the while-loop condition and the second being the tree representing the while-loop body.

This tree is called the concrete syntax tree since it is based on the specification of the actual concrete syntax, as opposed to a simplified abstract syntax that is sometimes used in the following semantic analysis phase.

Semantic analysis:-

This phases examines a tree representation of the program to perform analysis such as type checking and error reporting.

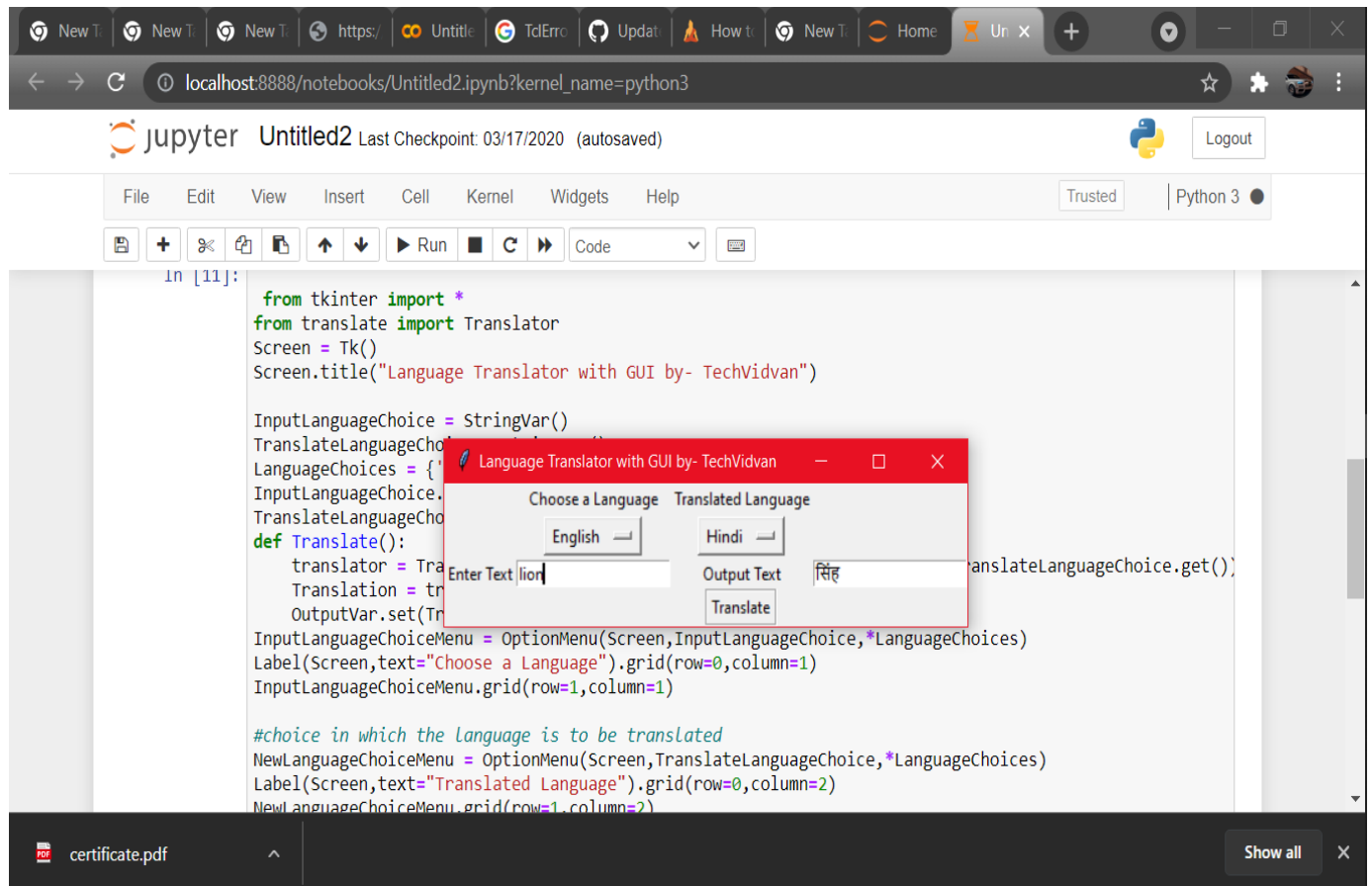
Code generation:-

This is the final phase in which some translation of the tree representation is generated. This may be byte-code, machine code, or a translation to a language such as C. In most compilers this phase is preceded by an optimization phase.

4- Experimental analysis:-

- Installing translate
- Importing translate and tkinter
- Initializing window
- Creating tuple for choosing languages
- Creating a function for translating the text
- Choice for input language and the language in which the text is to be translated
- Input and Output text

5-Output ScreenShot:-



6-Conclusion and Future scope:-

Translation turns a text of source language(SL) into a correct and understandable version of target language(TL) without losing the suggestion of the original. Many people think that being bilingual is all that is needed to be a translator.

Mastering artificial intelligence and deep learning will create a new generation of translation software. One that delivers more accurate versions of the original content, in more languages. The future of translation will cover more cultures, as the internet continues to penetrate in emerging countries worldwide.

7-Reference:-

<https://techvidvan.com/tutorials/python-language-translator/>