

Proyecto Translog

Área de Ingeniería en Computadores

Instituto Tecnológico de Costa Rica

CE310 - Lenguajes, Compiladores e Intérpretes

Estudiantes

Allan Josué Calderón Quirós - 2018114634

Ronny Josué Santamaria Vargas - 2018109283

Antony Fabián Fallas Elizondo - 2018178906

Profesor

Marco Rivera Meneses

I Semestre 2020

Manual de Usuario

El programa es una implementación sencilla de un traductor español-inglés. Este traductor toma en principio las oraciones en español o inglés con la estructura sujeto+verbo+predicado y las traduce al idioma objetivo. Para esto, utiliza gramáticas libres de contexto que realizan las equivalencias gramáticas de ambos idiomas. También acepta ciertas oraciones con otras estructuras, por ejemplo preguntas. Además, utiliza una base de datos con palabras en español y su equivalente en inglés. En caso de que la traducción no sea posible basado en las reglas definidas, pueden darse varios casos:

- La oración introducida no tiene sentido.
- Utilización de signos no reconocidos por el programa, tales como signos de exclamación y pregunta, tildes, ñ.
- La oración introducida utiliza formas no “formales” de expresión, tales como contracciones, uniones informales de palabras o gramaticalmente incorrectas. En el caso del español, las palabras tildadas o con “ñ” pueden llevar a errores.
- Al menos una de las palabras introducidas no está en la base de datos.
- Alguna de las palabras que conforman la oración tiene varios significados que dependen del contexto, como cayó del verbo caer y cayó del verbo callar o siento del verbo sentir y siento del verbo sentar.
- No se está utilizando ninguna de las gramáticas aceptadas por el programa.

Dependiendo del error que se obtenga, el usuario recibirá un mensaje de error indicando que la traducción no se podrá realizar, se recibirá una traducción errónea o no recibirá ningún mensaje, lo cual indicará que el programa se cayó.

Requisitos para ejecutar el programa:

Tener instalado el lenguaje de programación PROLOG. En caso de no tenerlo se puede descargar desde su página web <https://www.swi-prolog.org/download/stable>.

Una vez obtenido el código fuente, ejecutar el archivo TransLogEl.pl haciendo doble click sobre él.

Al principio se mostrarán advertencias de variables singleton, esto implica que al guardar una traducción en una variable esta variable será inmutable, por lo que para realizar otra traducción se debe correr de nuevo el programa.

Una vez dentro, se leerá un mensaje solicitando el lenguaje a traducir. Se debe escribir “EN”. para traducir de inglés a español, o “ES”. para traducir de español a inglés.

Para traducir una oración, se debe introducir la misma en minúsculas y entre comillas. Posteriormente, ingresar un punto , que significa el final de la oración. Solo se debe ingresar una oración a la vez.

En caso de que la traducción sea posible, se obtendrá una lista que contiene la oración traducida. Para traducir otra oración se debe ejecutar el archivo nuevamente.

Ejemplos:

```
?- transLog.  
Ingrese el idioma a traducir / Type the language to translate  
|: "EN".  
Type text to translate  
|: "the car is red".  
el carro es rojo  
true.
```

```
?- transLog.  
Ingrese el idioma a traducir / Type the language to translate  
|: "ES".  
Escriba el texto para traducir  
|: "el hombre es grande".  
the man is big  
true.
```

```
?- trasLog.  
Correct to: "transLog"? yes  
  
Ingrese el idioma a traducir / Type the language to translate  
|: |: "ES"  
|: .  
Escriba el texto para traducir  
|: "Elhombre come manzana".  
NO SE PUEDE TRADUCIR  
true.
```

1.Documentación

1.1. Descripción de las reglas implementadas:

-Las reglas que se implementaron brindan una estructura para dividir la oración en sintagmas verbal, nominal y adjetivo ya que estos son los más utilizados en las oraciones, esta estructura se hizo con el fin de reconocer una cantidad considerable de oraciones o frases que se puedan ingresar y traducir.

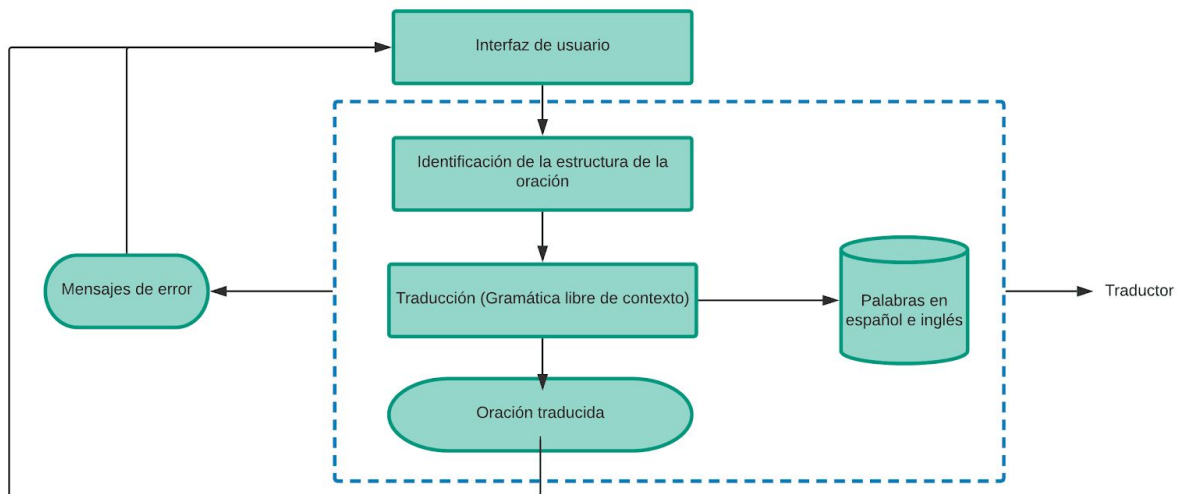
-Con lo que respecta a los hechos estos se implementaron tomando en cuenta la estructura que se le dio a la traducción de oraciones y poder dividirla en los sintagmas anteriormente mencionados, estos hechos poseen pronombres, determinantes, sustantivos, adjetivos y verbos, los cuales se conjugaron según tiempo (pasado, presente o futuro), persona (primera, segunda o tercera) y cantidad (plural o singular), para realizar una traducción más completa.

1.2.Descripción de las estructuras de datos desarrolladas:

-Listas:Una lista en Prolog es una colección de términos, que es útil para agrupar artículos, o para tratar con grandes volúmenes de datos relacionados, etc. Las listas están encerradas entre corchetes, y los artículos están separados por comas. La longitud de una lista es el número de los artículos que contiene.
Las listas permiten acceder y almacenar información.

1.3. Descripción detallada de los algoritmos desarrollados

-Se utilizó el algoritmo de concatenación de listas que toma dos listas agrega al final de la primer lista, la segunda lista. Esto para mostrar el resultado final en una lista con la traducción de las palabras que forman la oración.



1.4. Problemas sin solución:

-La implementación de un algoritmo de traducción 100% funcional es un trabajo increíblemente complejo que no ha sido resuelto por completo, aun por las grandes empresas como Google que han invertido muchos recursos en esta labor. Además de ser casi utópica la idea debido a todas las variaciones y dialectos que existen en un mismo lenguaje. Por esta razón, y considerando nuestra limitación en tiempo, recursos y conocimiento, la implementación de un traductor es una tarea muy complicada que solo se realizó de una manera básica y demostrativa. Por lo tanto, muchas oraciones (probablemente la mayoría de las oraciones que se pueden formar en ambas lenguas) no podrán ser traducidas correctamente por este algoritmo.

-El algoritmo muestra advertencias al ejecutar el programa, ya que las variables son inmutables, por lo que al seleccionar una traducción ya no se pueden reasignar las variables que se utilizan. Esto es debido al paradigma de programación utilizado, por lo que no se encontró forma de suprimir estas advertencias.

1.5. Plan de Actividades realizadas por estudiante:

División de las actividades por estudiante:

Encargado	Tiempo estimado	Actividad	Fecha de entrega
Allan Calderón	15 h	Implementación del traductor, documentación	21/07
Ronny Santamaría	16 h	Implementación del traductor, creación de base de datos	21/07
Fabian Fallas-Allan Calderón	15 h	Implementación del traductor, interfaz	21/07
Fabian Fallas-Allan Calderón-Ronny Santamaría	5 h	Debug, implementación y corrección de errores. Entrega del proyecto	22/07

1.6. Problemas encontrados:

-En algunas ocasiones se ingresaba una oración con la estructura descrita necesaria para la traducción y retornaba un error, esto se resolvió dando una estructura a los hechos primero se coloca la palabra en español y luego en inglés, esto precedido por los requerimientos previos que posee los hechos por ejemplo: verbo(cantidad, tiempo, persona, [palabra en español], [palabra en inglés]).

-Conforme aumenta el tamaño de la base de datos, el tiempo que tarda en realizarse la traducción aumenta considerablemente. Al incluir una base de más de 600 adjetivos, el traductor tomó más de 5 minutos en realizar la traducción. Por eso la base de datos que existe es demostrativa, ya que una base de datos exhaustiva toma demasiado tiempo en realizar la traducción. Y el caso anterior no tomó en cuenta la base de datos de verbos, por ejemplo.

- A pesar de los múltiples intentos la función para traducir negaciones en presente simple no se logra concluir esto debido a que mientras se realizaba el código se presentó diversos problemas; uno fue que el usuario deberá ingresar un sujeto y se debía reconocer que tipo de sujeto era, el cual el código creado no lograba reconocer, además de que el código generaba un error al cual no se le logra encontrar solución, este error no permite reconocer si era o no una negación por lo cual el código no cumplía su función.

-Si alguna de los elementos de la oración no está explícito no se logra hacer la traducción.

1.7. Conclusiones y Recomendaciones del proyecto:

-La creación de un traductor es un proceso muy complejo, tanto por su desarrollo algorítmico como por su sustento teórico, ya que existen diferentes dialectos entre los lenguajes, diferentes conjugaciones, estructuras gramaticales, signos, tiempos, expresiones, palabras aceptadas sólo en ciertas regiones etc. Esta es la razón de porque grandes traductores como el traductor de Google o Deepl aún presentan errores y dificultades para traducir.

-La forma de programar en Prolog, debido a su paradigma, es muy distinta a la de otros lenguajes de programación, es por esto que al aprender un lenguaje se debe estar abierto a nuevas reglas y formas de programar que no existen en otros lenguajes. Esta forma de pensar es útil al aprender un nuevo lenguaje, ya que no limita solo a lo que ya se conoce.

-Al desarrollar un algoritmo traductor es recomendable utilizar los mejores algoritmos de búsqueda posibles, ya que una búsqueda lineal entre las miles de palabras que existen en un lenguaje es un proceso que toma demasiado tiempo.

1.8. Bibliografía consultada en todo el proyecto:

Monferrer, T. E., Toledo, F., & Pacheco, J. (2001). El lenguaje de programación Prolog. Valencia.

Monera, V., & Monera, V. (2020). LA ORACIÓN. Estructura y tipos. Retrieved 6 July 2020, from <https://www.victoriamonera.com/la-oracion-estructura-y-tipos/>

SWI-Prolog . (2020).Reference manual. Retrieved 20 July 2020, from <https://www.swi-prolog.org/>

Estructuras básicas de oraciones en el idioma inglés. (2020). Retrieved 20 July 2020, from <https://www.really-learn-english.com/sentence-structure-es.html>

Questions in the Simple Present, Questions with do, does. (2020). Retrieved 16 July 2020, from <https://www.english-hilfen.de/en/grammar/frage4.htm>

2.Bitácoras digitales

Allan Calderón

Fecha	Tiempo	Actividad
14/07	1 h	Lectura de los requerimientos del programa, organización y división del trabajo
16/07	1.5 h	Análisis de código, investigación de la implementación de las estructuras sintagma nominal+sintagma verbal en presente simple
18/07	3 h	Trabajo en la implementación de reglas gramaticales, principalmente considerando preguntas en presente simple
20/07	3 h	Trabajo en implementación de excepciones, preguntas. Corrección de base de datos
21/07	1 h	Documentación, mejora en interfaz (mostrar

		traducciones como strings en vez de listas)
22/07	2.5 h	Revisión de errores, intento de implementación de excepciones, entrega del proyecto

Ronny Santamaria

Fecha	Tiempo	Actividad
06/07/2020	2 h	Investigación de la estructura que poseen las oraciones en ingles y en español. Formulación de la estructura que tendrán los hechos de la base de datos.
09/07/2020	2 h	Implementación de traducciones a la base de datos.
10/07/2020	2 h	Creación de traducciones monosilábicas. Pruebas al código creado.
12/07/2020	6 h	Definición y creación de la estructura que tendrán las oraciones en el traductor. Pruebas al traductor con la estructura creada.
15/07/2020	1 h	Se hicieron pruebas agregando traducciones a la base de datos.
20/07/2020	2 h	Creación de la estructura para hacer la traducción de algunas preguntas y excepciones.

Fabian Fallas

Fecha	Tiempo	Actividad
10/07/2020	1 h 30 min	Investigación sobre cómo se forman las estructuras en inglés para empezar a implementar el traductor
14/07/2020	2 h	Análisis del código, búsqueda sobre las leyes gramaticales para excepciones
16/07/2020	5 h	Se comienza a trabajar en la implementación de las negaciones en presente simple

18/07/2020	3 h	Se continua la implementación de negaciones en presente simple
19/07/2020	1 h	Pruebas con el código en busca de errores
21/07/2020	1 h	Documentacion, corrección de errores en el código