Universidad Nacional de Rosario

Facultad de Ciencias Exactas,
Ingeniería y Agrimensura

Trabajo Final

Análisis de Lenguajes de Programación

Antonio Locascio
L-2876/2

Febrero 2018

Índice

1 Presentación del proyecto

1.1 Introducción

El objetivo de este trabajo es desarrollar una herramienta que permita analizar formularios ópticamente. Esto consiste en reconocer ciertas características presentes en una imagen de un documento e interpretarlas. De esta forma, es posible obtener rápidamente las respuestas a un formulario que un usuario completó en formato papel.

1.2 Motivación

La idea de este proyecto surge del contexto de debate sobre el voto electrónico en Argentina. Dados los riesgos que emanan de la introducción de sistemas electrónicos en la fase de emisión del voto, una alternativa comunmente propuesta es la utilización de tecnología como auxilio para el conteo de los sufragios.[1]

En este informe se presenta una implementación de un sistema que realiza esta tarea en un contexto acotado.

1.3 Tipo de documentos

La clase de documentos con los que se puede utilizar esta herramienta es la de aquellos que corresponden a formularios de opción múltiple que deben ser completados en papel. Para poder ser reconocido correctamente, un documento debe tener ciertos atributos visuales detallados en la **Sección 2.1**.

En la **Sección 3**, se introduce un DSL diseñado para describir estos formularios. Estas descripciones se utilizarán tanto para la creación de los documentos, como para el análisis óptico posterior.

1.4 Usos

Además del uso mencionado anteriormente (conteo de votos), este proyecto puede resultar útil para el relevamiento de encuestas y la corrección de exámenes.

2 Utilidades y especificaciones

El sistema cuenta con dos utilidades principales que se describen a continuación.

2.1 Generación del documento (`make`)

Utilizada para crear un documento PDF en base a una descripción del formulario. Este sigue las siguientes reglas de formato:

- Cada sección se presenta dentro de un rectángulo de contorno negro.
- Una sección puede contener subsecciones o un conjunto de opciones.
- Cada opción debe contar con un círculo de contorno negro, donde se marcaría de elegirse.
- Toda página debe tener una sola sección principal que encapsula al resto.

Si bien el resultado de esta utilidad puede utilizarse directamente, su función fundamental es crear una guía de diseño para el usuario. Por esta razón, no se hace mayor hincapié en el sentido estético del diseño del documento, que está por fuera del alcance de este trabajo.

2.2 Escaneo del documento (`scan`)

Una vez que se cuenta con el formulario, este se imprime y completa. Para esto último, se debe rellenar con un color oscuro los círculos correspondientes a las opciones que se desean elegir. No es necesario que se cubra la totalidad del círculo.

El paso siguiente es escanear (digitalizar) los documentos. Para un óptimo funcionamiento, las hojas deben estar escaneadas en una resolución de al menos 300 ppi (2480 X 3508 píxeles para A4), en formato `jpg`.

Como salida se presentan los resultados encontrados. Su formato se explica en la **Sección 4.2**.

3 DSL para la descripción de documentos

La definición de un lenguaje de dominio específico que describa los documentos es necesaria para ambas utilidades del proyecto. En un primer lugar, permite especificar formalmente el contenido y formato del formulario para su generación. En cuanto a la segunda, se utiliza para comparar e interpretar el resultado del reconocimiento óptico.

Definición formal

A continuación se presenta la gramática de sintáxis concreta en BNF:

$$
\begin{aligned}
DOC &::= \textit{page}\ SECT\ \textit{endpage}\ DOC \mid \lambda \\
SECT &::= \textit{section}\ TITLE\ CONT\ \textit{endsection} \\
CONT &::= \textit{section}\ TITLE\ CONT\ \textit{endsection}\ SUBS \mid OPTS_1 \\
SUBS &::= \textit{section}\ TITLE\ CONT\ \textit{endsection}\ SUBS \mid \lambda \\
OPTS_1 &::= \textit{option}\ TITLE\ OPTS \\
OPTS &::= \textit{option}\ TITLE\ OPTS \mid \textit{multiple}\ RES \mid \lambda \\
RES &::= \textit{yes} \mid \textit{no} \\
TITLE &::= \text{“}\ STR\ \text{”} \\
STR &::= CHAR\ STR \mid \lambda
\end{aligned}
$$

Se utiliza $CHAR$ como el conjunto de caracteres alfanuméricos y de puntuación. En esta gramática se especifica que un documento puede tener más de una página y que cada una de ellas tiene una sola sección principal. Además, el contenido de una sección puede ser subsecciones u opciones, pero no ambas a la vez. Por último, una sección de opciones puede incluir al final una restricción para la selección de múltiples respuestas. Por omisión esto está permitido.

Un ejemplo de un documento puede encontrarse en el **Anexo**.

4 Decisiones de diseño

4.1 Tipos de datos

En esta sección se explicarán las funciones de los distintos tipos de datos utilizados en el proyecto. Las definiciones e instancias de estos se encuentran en el archivo `Types.hs`.

El primer tipo a detallar es `Structure`, que se define de la siguiente manera:

```
type Structure = [StructPage]
data StructPage = Rectangle  [StructPage] | Circle Bool
```

Este tipo de datos sirve para representar la estructura reconocida por la herramienta luego de analizar un documento escaneado. Como indica la definición, la estructura de un formulario consiste de una lista de páginas, que a su vez contienen rectángulos con subestructuras y círculos marcados o no.

```
type Document = [Page]
data Page = Section Title Subsection
data Subsection = Subs [Page] | Options Restriction [Option]
```

El tipo `Document` se utiliza para la representación interna de la descripción de un documento. Corresponde a la sintáxis abstracta del lenguaje definido en la sección anterior.

Por su parte, los tipos `Punto` y `Contorno` solo se definen con el objetivo de simplificar el uso de la librería OpenCV.

```
type Result = [PageResult]
data PageResult = Sect [PageResult] | Ans [Res]
type Res = Bool
```

Por último, este tipo de datos se utiliza para representar los resultados de la evaluación final de la utilidad `scan`. Para esto, se usa una estructura recursiva similar a la del documento, en la que las respuestas se guardan como listas de booleanos que indican si una opción fue marcada.

4.2 Formato de salida

Además del tipo de datos para los resultados recién explicado, se define el siguiente tipo:

```
type FlatResult = [(SectNum, [Int])]
type SectNum = [Int]
```

Este tipo de datos sirve para facilitar la lectura de los resultados. En particular, el tipo `SectNum` es una representación de los números de sección. Estos se interpretan como códigos punteados en los que la posición (de izquierda a derecha) indica mayor profundidad y cada número da un orden dentro de su nivel. Los números de sección permiten aplanar los resultados, manteniendo la información de la estructura del documento. La segunda componente de las tuplas indican los índices (iniciando en 1) de las opciones que se eligieron en cada pregunta.

Como se muestra en la **Sección 6.2**, este formato corresponde a la salida impresa en pantalla cuando se ejecuta el comando `scan`.

5 Organización del código

Para desarrollar esta herramienta, se utilizó el lenguaje de programación Haskell y la plataforma Stack. La estructuración del código es la siguiente:

```
TPFinalALP
├── LICENSE
├── README.md
├── Setup.hs
├── stack.yaml
├── src
│   ├── BlurImage.hs
│   ├── Eval.hs
│   ├── ImageRec.hs
│   ├── Parser.hs
│   ├── PDFMaker.hs
│   ├── Threshold.hs
│   └── Types.hs
├── app
│   └── Main.hs
└── test
    ├── sampleDescr
    ├── sampleDocs
    └── sampleScans
```

5.1 Módulos

A continuación se describen brevemente los aspectos fundamentales de cada módulo. Descripciones más detalladas de cada función pueden encontrarse en forma de comentarios en los archivos de código.

`PDFMaker` define las funciones de creación de un archivo `PDF` en base a la descripción de un documento. Los detalles de esta tarea se encuentran en la **Sección 2.1**.

El módulo `ImageRec` se encarga del reconocimiento óptico de una imagen de un documento. Expone una sola función, `scanImage`, que toma la ruta de la imagen y devuelve algo del tipo `IO (Either Error StructPage)`, indicando que se realizan operaciones de entrada/salida y que pueden ocurrir errores.

`ImageRec` cuenta con dos módulos auxiliares: `BlurImage` y `Threshold`. Estos implementan funciones para difuminar una imagen (para minimizar el efecto del ruido) y para binarizarla. Esto último consiste en hacer que cada píxel sea blanco o negro, dependiendo de si su intensidad original supera un cierto límite y sirve para simplificar el análisis de la imagen.

Las funciones que se ocupan del parseo del lenguaje de descripción de documentos a un elemento del tipo `Document` se encuentran en el módulo `Parser`. Como se mencionó en la **Sección 4.1**, `Types` se ocupa de definir los tipos de datos y dar sus instancias.

Por último, el módulo `Eval` expone dos funciones. La primera, `eval`, dados un documento y una estructura escaneada las evalúa para verificar que coincidan. La segunda, `flattenResult`, aplana los resultados al formato punteado explicado anteriormente.

6 Manual de uso

6.1 Guía de instalación

Previo a la instalación de la herramienta, es necesario contar con:

- Stack: https://docs.haskellstack.org/en/stable/README/
- OpenCV: https://opencv.org/
- Código Fuente: https://github.com/antoniolocascio/TPFinalALP

Contando con lo anterior, la instalación del sistema consiste en simplemente ejecutar los siguientes comandos desde una terminal, trabajando desde la carpeta `TPFinalALP`.

```
$ stack setup
$ stack build
```

6.2 Ejemplo de uso

Se presenta un ejemplo completo de uso para ambas utilidades. Para comenzar, se cuenta con una descripción de un documento en el DSL definido para el proyecto. Este se encuentra en `TPFinalALP/test/sampleDescr/sample1.txt`, y se adjunta a este informe en el **Anexo**.

El formulario cuenta con dos páginas. En la primera se describen cuatro subsecciones, cada una correspondiente a una pregunta. La segunda página está compuesta de dos subsecciones que cuentan con dos secciones de pregunta cada una.

En primer lugar se muestra la función `make`, cuya sintáxis es:

```
make DOC_DESCR_PATH OUTPUT_PATH
```

Continuando con el ejemplo, el comando resultante es:

```
$ stack exec tp make test/sampleDescr/sample1.txt
  test/sampleDocs/sample1.pdf
```

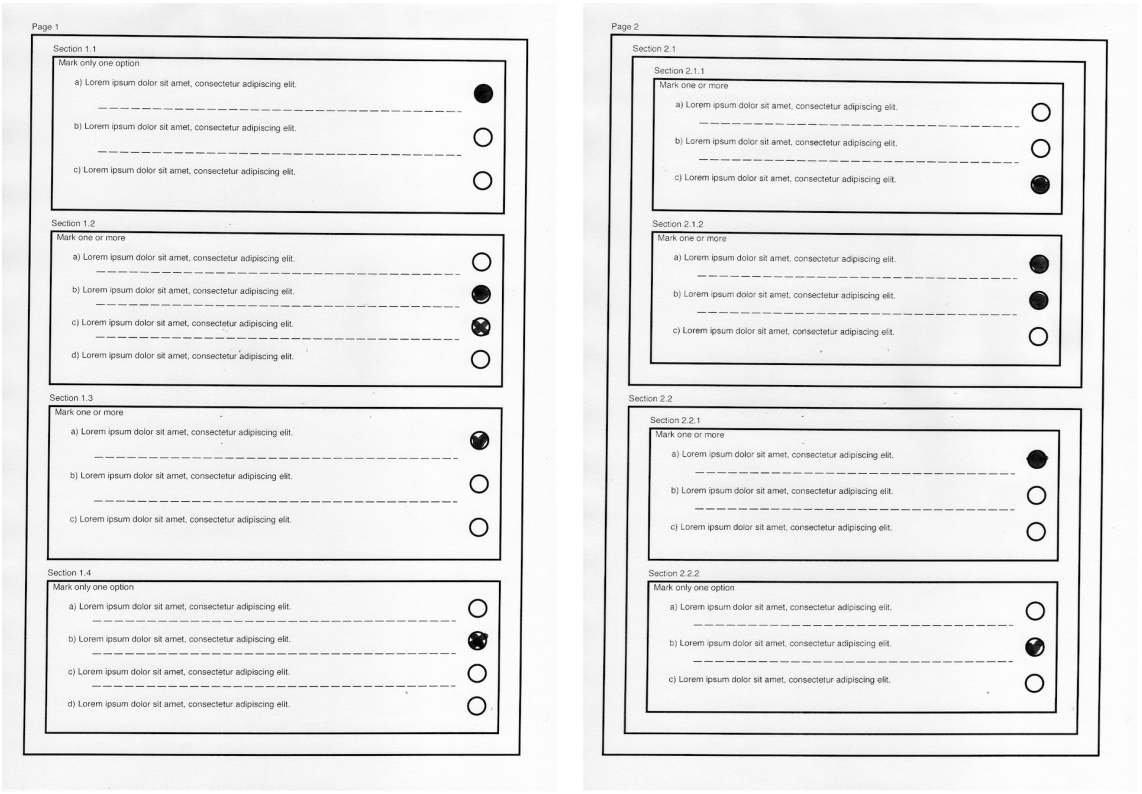
Esto genera el documento `sample1.pdf`, también disponible en el **Anexo**, que para seguir con esta demostración se imprime, completa en papel y digitaliza. Como resultado, se obtienen las imágenes `scan1_1.jpg` y `scan1_2.jpg` que se encuentran en `TPFinalALP/test/sampleScans`.

La sintáxis de la segunda función, `scan`, tiene la forma:

```
scan DOC_DESCRIPTION_FILEPATH IMAGE_FILEPATH(S)
```

Aplicándola al ejemplo, el comando correspondiente es:

```
$ stack exec tp scan test/sampleDescr/sample1.txt
  test/sampleScans/scan1_1.jpg test/sampleScans/scan1_2.jpg
```



Formulario digitalizado: `scan1_1.jpg` y `scan1_2.jpg`

Al ejecutar este comando, se imprimen en pantalla los resultados en el formato aplanado explicado en la **Sección 4.2**. Para este caso, la salida es:

```
Results:
1.1: [1]
1.2: [2,3]
1.3: [1]
1.4: [2]
2.1.1: [3]
2.1.2: [1,2]
2.2.1: [1]
2.2.2: [2]
```

7 Bibliografía y recursos

[1] CONICET. *Análisis de factibilidad en la implementación de tecnología en diferentes aspectos y etapas del proceso electoral*. URL: http://www.conicet.gov.ar/wp-content/uploads/Analisis_factibilidad_implementacion_tecnologia_proceso_electoral.pdf. (consulta: ene. 2018).

[2] *Graphics.PDF.Documentation*. URL: https://hackage.haskell.org/package/HPDF-1.4.10/docs/Graphics-PDF-Documentation.html. (consulta: ene. 2018).

[3] *Respuesta en StackOverflow*. URL: https://stackoverflow.com/questions/39661287/gaussianblurimage-in-haskell-opencv-haskell-binding-to-opencv-3-1. (consulta: ene. 2018).

[4] Adrian Rosebrock. *OpenCV shape detection*. URL: https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/. (consulta: ene. 2018).

8 Anexo

8.1 sample1.txt

```
page
  section "Page 1"
    section "Section 1.1"
      option "a) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      option "b) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      option "c) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      multiple no
    endsection

    section "Section 1.2"
      option "a) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      option "b) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      option "c) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      option "d) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
    endsection

    section "Section 1.3"
      option "a) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      option "b) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      option "c) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
    endsection

    section "Section 1.4"
      option "a) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      option "b) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      option "c) Lorem ipsum dolor sit amet, consectetur
```

```
   adipiscing elit."
      option "d) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      multiple no
    endsection

  endsection
endpage

page
  section "Page 2"
    section "Section 2.1"
      section "Section 2.1.1"
        option "a) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
        option "b) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
        option "c) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
        multiple yes
      endsection

      section "Section 2.1.2"
        option "a) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
        option "b) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
        option "c) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      endsection
    endsection

    section "Section 2.2"
      section "Section 2.2.1"
        option "a) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
        option "b) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
        option "c) Lorem ipsum dolor sit amet, consectetur
   adipiscing elit."
      endsection

      section "Section 2.2.2"
```

```
        option "a) Lorem ipsum dolor sit amet, consectetur
    adipiscing elit."
        option "b) Lorem ipsum dolor sit amet, consectetur
    adipiscing elit."
        option "c) Lorem ipsum dolor sit amet, consectetur
    adipiscing elit."
        multiple no
      endsection
     endsection

  endsection
endpage
```

8.2 sample1.pdf

Se adjunta en las páginas siguientes.

Section 1.1

Mark only one option

a) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

b) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

c) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Section 1.2

Mark one or more

a) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

b) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

c) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

d) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Section 1.3

Mark one or more

a) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

b) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

c) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Section 1.4

Mark only one option

a) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

b) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

c) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

d) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Section 2.1

Section 2.1.1

Mark one or more

a) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

b) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

c) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Section 2.1.2

Mark one or more

a) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

b) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

c) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Section 2.2

Section 2.2.1

Mark one or more

a) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

b) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

c) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Section 2.2.2

Mark only one option

a) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

b) Lorem ipsum dolor sit amet, consectetur adipiscing elit.

c) Lorem ipsum dolor sit amet, consectetur adipiscing elit.