UNIVERSIDAD DE SONORA

División de Ciencias Exactas y Naturales

Departamento de Matemáticas

Programa de Maestría en Ciencias con Especialidad en Matemática Educativa

Estudio sobre el Aprendizaje de las Leyes de De Morgan desde la Teoría APOE

Tesis que para obtener el grado de:

Maestría en Ciencias con Especialidad en Matemática Educativa

Presenta:

Jesús Rafael Noriega Mendoza

Director de Tesis

Dr. César Fabián Romero Félix

Hermosillo, Sonora, octubre de 2018.

Hermosillo, Sonora. 19 de octubre de 2018

Asunto: Cesión de derechos

Universidad de Sonora

Presente

Por este conducto hago constar que soy autor y titular de la obra denominada *Estudio sobre el Aprendizaje de las Leyes de De Morgan desde la Teoría APOE*, en los sucesivo LA OBRA, realizada como trabajo terminal con el propósito de obtener el Grado de Maestría en Ciencias con Especialidad en Matemática Educativa, en virtud de lo cual autorizo a la Universidad de Sonora (UNISON) para que efectúe la divulgación, publicación, comunicación pública, distribución publica, distribución electrónica y reproducción, así como la digitalización de la misma, con fines académicos o propios de la institución y se integre a los repositorios de la universidad, estatales, regionales, nacionales e internacionales.

La UNISON se compromete a respetar en todo momento mi autoría y a otorgarme el crédito correspondiente en todas las actividades mencionadas anteriormente.

De la misma manera, manifiesto que el contenido académico, literario, la edición y en general cualquier parte de LA OBRA son de mi entera responsabilidad, por lo que deslindo a la UNISON por cualquier violación de los derechos de autor y/o propiedad intelectual y/o cualquier responsabilidad relacionada con la OBRA que cometa el suscrito frente a terceros.

Atentamente

____

Jesús Rafael Noriega Mendoza

*Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo brindado para mi formación, con la beca de manutención **611946***

Contenido

Introducción 1

1 Antecedentes y Problemática 4

1.1 Justificación 6

1.2 Dificultades en aprendizaje de Programación relacionadas con leyes de De Morgan 6

1.3 Inserción curricular 10

1.4 Objetivos de investigación 11

1.5 Preguntas de investigación 12

2 Marco Teórico 13

2.1 La abstracción reflexiva 13

2.2 Las construcciones mentales 14

2.2.1 Acción 14

2.2.2 Proceso 14

2.2.3 Objeto 14

2.2.4 Esquema 15

2.3 Descomposición Genética 15

3 Metodología 17

3.1 Análisis teórico 17

3.2 Diseño e implementación de la enseñanza: 17

3.3 Observación, Análisis y verificación de los datos: 18

4 Diseño y análisis a priori de actividades exploratorias 21

4.1 Diseño de problemas y respuestas esperadas 21

4.1.1 Actividad 0 22

4.1.2 Actividad 1 24

4.1.3 Actividad 2 32

4.2 Análisis a priori bajo al marco APOE 36

4.2.1 Actividad 0 36

4.2.2 Actividad 1 37

4.2.3 Actividad 2. 42
5 Análisis de datos y presentación de resultados 47
5.1 Implementación de las actividades 47
5.2 Análisis de las concepciones de las leyes de De Morgan 47
5.2.1 Acción Primera Ley 48
5.2.2 Acción Segunda Ley 54
5.2.3 Procesos para las leyes de De Morgan 61
5.2.4 Procesos generalizados 68
6 Conclusiones 73
6.1 Sobre las concepciones buscadas 73
6.1.1 Resultados de la concepción Acción 73
6.1.2 Resultados de la concepción Proceso 73
6.1.3 Resultados de la concepción Objeto 74
6.1.4 Procesos avanzados 74
6.1.5 Construcciones previas 74
6.1.6 Complejidad de los problemas de programación 74
6.2 Reflexiones finales 75
6.2.1 Descomposición genética preliminar 75
6.2.2 Versión general de la descomposición genética preliminar 79
6.2.3 Implicaciones didácticas 80
6.2.4 Sobre el alcance de los objetivos 81
6.2.5 Sobre la problemática general 82
6.2.6 Sobre la utilidad de estos resultados para futuros trabajos 82
6.2.7 Sobre problemas abiertos 83
Referencias 85
Anexos 86
Actividad 0. 87
Actividad 1. 90

Actividad 2. 103

Introducción

Los orígenes de la lógica se le atribuye al pensador griego Aristóteles alrededor de los años 200 y 350 antes de la era común, aunque, hay indicios de que pudo dotar aún más tiempo atrás, el tipo de lógica que implementaban los griegos se fundaba particularmente en la argumentación deductiva en donde la mescla de la lingüística, la filosofía y las matemáticas hacían presencia, es por ello que no se le reconocía como una rama de las matemáticas formales tales como la geometría o la aritmética. Fue hasta el siglo XVII donde se empezó a desarrollar un cambio en el cual contemporáneos de aquellas épocas como Descartes y Leibniz empezaron a retomar las ideas de los griegos y a estas darles una simbología con el propósito de desarrollar un lenguaje universal del razonamiento, mismo que no se pudo lograr es su tiempo. Esas ideas tardaron en establecerse y hasta 1850 fueron retomadas por George Boole quien algebraizó la lógica aristotélica y propició el nacimiento de la lógica como rama de las matemáticas.

George Boole en conjunto con Augustus De Morgan trabajaron en darle solidez al razonamiento aristotélico trabajando específicamente en los silogismos y las inferencias de proposiciones, a partir de este punto hubo gran avance y la lógica dio paso a la creación de nuevos campos de la matemática algunos como el álgebra abstracta y la teoría de conjuntos.

Con los avances que se tenían de la lógica matemática y también del lado tecnológico se empezó a desarrollar los primeros computadores entre 1940 y 1950, las cuales seguían una estructura lógico-matemática para su funcionamiento, posteriormente surgieron otras generaciones y cada vez con más características, pero siempre partiendo de las bases de la lógica.

Como vemos los pilares fundamentales en los que descansa la Matemática proviene directamente de la lógica, la cual a su vez surge de la filosofía que se encarga de estudiar los principios generales del conocimiento, así como la estructura del pensamiento. El estudio de la lógica como hoy la conocemos se divide en diferentes categorías, es así como existen diferentes tipos de lógicas, entre ellas tenemos; lógica simbólica, lógica Matemática, lógica informal, lógica modal, lógica clásica, lógica no clásica y por último la lógica formal.

La lógica formal se encarga de estudiar las proposiciones, argumentos y afirmación de oraciones desde el punto de vista estructural, de igual manera son importantes los demás tipos de lógica, sin embargo, nuestro trabajo está solamente dirigido a estudiantes en donde su enfoque está dirigido a la lógica Matemática y que a su vez dentro de esta se encuentra la lógica proposicional.

Observamos que, al menos en la Universidad de Sonora, no todas las ramas de Ciencias Exactas y sociales contemplan en sus planes de estudio asignaturas relacionadas con temas de lógica. Aun así, en algunos casos han permanecido materias de este tipo en programas de carreras universitarias que tienen un carácter de formación en informática, materias como: lenguajes de programación, base de datos, diseño de algoritmos entre otras asignaturas. La lógica es sin duda uno de los fundamentos que proporcionan la madurez necesaria para asimilar los conceptos de las materias mencionadas, pues generalmente se requiere un análisis de las propiedades lógicas antes de diseñar o desarrollar algún sistema informático o algoritmo.

Una de las motivaciones a realizar este trabajo fue la falta de resultados que existen en relación con los temas de lógica en Matemática educativa, siendo esta rama de gran relevancia en el campo de las Ciencias Exactas debido a que fundamenta métodos muy útiles en las carreras de programación y además es una de las leyes básicas de la lógica booleana, así como en el ámbito social. Dada la importancia de su estudio, desde el punto de vista cognitivo, prevemos que analizar temas fundamentales de lógica nos podría llevar a determinar algunas inconsistencias y dificultades en su aprendizaje. De tal manera, en el presente trabajo mostramos un primer estudio sobre el aprendizaje de las leyes de De Morgan desde la teoría APOE, en el cual se evalúan dos aspectos, el diseño de actividades de evaluación y el desarrollo de las estructuras mentales asociadas a este concepto.

A continuación, mostraremos un contenido breve correspondiente a cada capítulo:

Capítulo 1: Antecedentes y problemática

En este primer capítulo daremos una breve explicación de las nociones y la importancia que ha tenido la lógica para el desarrollo de todas las ramas de la ciencia, así como las aplicaciones fuertes que se sustentan en el desarrollo de los sistemas tecnológicos, por consiguiente, nos enfocaremos a ciertos problemas que han sido desatendidos, mismos que están desde el punto de vista cognitivo. Además, presentaremos los alcances de este proyecto y lo que se propondrá presentar en los subsiguientes apartados.

Capítulo 2: Marco teórico

En este apartado expondremos las características y definiciones de los elementos teóricos que empleamos para los fines de este trabajo, provenientes del marco teórico APOE, por lo que se muestra una visión general de la estructura de este enfoque teórico.

Capítulo 3: Metodología

Aquí presentaremos una explicación general del ciclo de investigación propuesto en el paradigma de APOE, dando detalles de los elementos que lo conforman, así como una explicación en detalle sobre las tareas desempeñadas en cada etapa del ciclo.

Capítulo 4: Diseño y análisis a priori de actividades exploratorias

Daremos algunos argumentos y referencias que tomamos como base para efectuar el diseño de las actividades, posteriormente se mostrará un análisis a priori tanto de las respuestas esperadas como desde el punto vista del marco teórico.

Capítulo 5: Análisis de datos y presentación resultados

Se expondrán las respuestas de algunos estudiantes y se hará un análisis tomando como base el marco teórico para identificar y definir las concepciones mentales de los estudiantes y las relaciones entre estas; al comparar el comportamiento observado con las características que propusimos en el apartado anterior de las respuestas esperadas.

Capítulo 6: Conclusiones

En este último apartado extendemos los resultados que obtuvimos a partir de las concepciones que esperábamos obtener y como producto final del análisis expondremos la realización de una Descomposición Genética Preliminar; además de exponer brevemente algunos problemas abiertos.

1 Antecedentes y Problemática

Como sabemos, la lógica estudia las formas del razonamiento, es por ello que la enseñanza de la matemática excluyendo estos principios puede llegar a ser una tarea ardua o casi imposible de realizar; especialmente si lo que se pretende es tener una sólida formación de las ideas que están detrás de los conceptos y las definiciones matemáticas. Dentro de la Universidad de Sonora los temas correspondientes al área de la lógica Moderna la cual se presenta en algunos planes de estudio contempla: la lógica simbólica, enunciados condicionales, demostración por tablas de verdad, los silogismos entere otros tópicos; son ofrecidos a estudiantes que están cursando carreras que tienen como fin el desarrollo del área de computación y la electrónica, dejando fuera a otro tipo de ramas dentro de las carreras con el enfoque científico y del área de las ciencias exactas, esto puede corresponder a que dentro de sus actividades están en constante trabajo con las herramientas de la lógica.

El primer acercamiento con las propiedades básicas de la lógica no es algo trivial pues visto desde la forma general de las reglas de la lógica como los son las reglas primitivas, simbolización y de remplazo comprenden los fundamentos del razonamiento y estas no siempre son estudiadas en niveles tempranos en la formación del estudiante particularmente en los cursos que son preuniversitarios, es por ello que existen algunas complicaciones cuando se estudian por primera vez estos temas.

La Lógica está fundamentada por los siguientes principios: principio de identidad, principio de contradicción y principio del tercer excluido (Metafísica Libros IV & XI, Aristóteles); con los cuales se puede construir toda la estructura del razonamiento. Para ello, se utiliza un sistema de reglas que rigen la operatividad de la lógica de primer orden, algunas de ellas son: regla de conjunción y de disyunción, regla de condicional, regla de negación, reglas de condicional y conjunción, reglas de condicional y negación, reglas de De Morgan, reglas de equivalencia de conectivas, entre otras. Estas reglas son empleadas tanto en lenguajes naturales como en lenguajes formales, algunos ejemplos de lenguajes formales son los lenguajes informáticos y que están compuestos de variables, condicionales, bucles y funciones cuyas aplicaciones incluyen todos los programas a nivel sintaxis.

De esta manera, el lenguaje natural formado por la lengua hablada y escrita también está constituido de reglas. En este lenguaje las reglas de simbolización comprenden una herramienta muy útil al momento de crear oraciones con el fin de evitar ambigüedades y una falta de contextualización en entornos semánticos del habla cotidiana. Sin embargo, se han realizado investigaciones en el ámbito de la Psicología, en ambientes escolares y no escolares, en las cuales se ha demostrado en el lenguaje natural la existencia de algunas dificultades para utilizar o interpretar los conectivos lógicos tales como las conjunciones y las disyunciones, así como sus respectivas negaciones por parte de individuos con poca o nula formación en Lógica (Vest, 1981; Macbeth, Razumiejczyk & Campitelli, 2013). Los resultados de estas investigaciones indican que trabajar con las leyes de De Morgan es en una tarea ardua y complicada, sin importar si se está trabajando a nivel de lenguaje natural o en expresiones de carácter abstracto matemático.

Al respecto, los autores Macbeth, Razumiejczyk y Campitelli (2013) reportan dos estudios sobre el tema. En un primer estudio se seleccionaron 79 estudiantes del área de ciencias sociales, a los que se les presentaron oraciones escritas en lenguaje natural que involucraban las leyes de De Morgan; es decir, dado un enunciado, elegir de una lista la negación de éste. Los resultados obtenidos en este estudio muestran que más de 74% de los participantes tuvieron dificultades con los reactivos propuestos, es decir que tan solo 20 individuos pudieron contestar dos reactivos correctamente. Para el segundo estudio siguieron la misma estructura, solo que para este caso se emplearon reactivos que contemplaban proposiciones abstractas, incluyendo en lenguaje matemático; así mismo, se seleccionaron nuevos participantes, y se incluyeron temas matemáticos como números, ángulos y figuras geométricas. Los resultados de este estudio muestran que de los 86 participantes tan solo 18 tuvieron dos aciertos en los reactivos es decir que el 79% de los participantes presentaron una dificultad mayor en estos activos que en los del primer estudio. Estos resultados nos proporcionan una evidencia de la existencia de dificultades que son muy cercanas a las dificultades que estamos contemplando, pues se presenta que manejar negaciones de conjunciones y negaciones es complicado sin importar el contexto en el que se estén implementando.

Por otro lado, diversos autores postulan que, al tratar de negar oraciones, lo cual implica emplear las leyes de De Morgan, para distintas personas negar una conjunción (Primera Ley) es más o menos difícil que negar una disyunción (Segunda Ley); es decir, las leyes de De Morgan parecen ser desarrolladas de manera independiente, lo que permite tener mayor destreza con una que con la otra (ver por ejemplo, Khemlani et al., 2012; Macbeth, Razumiejczyk & Campitelli, 2013).

Si bien, los individuos que participaron en los mencionados estudios (Razumiejczyk & Campitelli, 2013) no son pertenecientes al campo de las ciencias exactas, es notable la dificultad al emplear los operadores lógicos en expresiones que se usan frecuentemente en el lenguaje natural y que esto les puede generar problemas en su vida diaria. Por la flexibilidad de los lenguajes naturales, podemos suponer que tales resultados no son tan graves como lo serían si se presentaran en usuarios de lenguajes formales como los de informática; así mismo, esperaríamos que los usuarios de lenguajes formales no cometieran estos errores. En el desarrollo del software, por ejemplo, es muy común que se encuentren errores a nivel de código, sobre todo en el aspecto lógico, lo cual puede conllevar a grandes catástrofes de programación, algunos casos ampliamente documentados son: NASA Mariner

I, sonda de Venus la cual no logró abandonar la atmosfera terrestre con lo que se mandó una orden para su autodestrucción, la causa de esta catástrofe se debió a un “punto” en vez de “coma” en un bucle “do” en el lenguaje de FORTRAN en 1962; el fallo del servicio de larga distancia de AT&T durante nueve horas a causa de una *sentencia break* incorrecta en lenguaje *C* en 1990 (Neumann, P.G 1995). Este tipo de errores es presentado tanto por profesionales experimentados, así como por programadores y estudiantes novatos, es por ello que la formación entre estudiantes de recién ingreso a carreras relacionadas a áreas de lógica y programación el aprendizaje correcto del uso de estas expresiones es de gran importancia.

1.1 Justificación

Uno de los objetivos iniciales que se contemplaban en este trabajo fue elaborar actividades para la enseñanza de los temas conjuntistas enfocados a áreas de las ciencias de la computación debido a su gran utilidad como herramienta en diversas áreas, por ello se pensó en actividades que fueran acorde a los temas que los estudiantes de computación tomaban en sus cursos tales como: *Base de datos*, *Teoría de códigos*, *Matemáticas discreta* entre otros. Consiguiente se distinguió en el desarrollo de la parte lógica en la formación de un estudiante, pues esto le aporta un buen sustento en tareas y ramas dentro de la computación. En cuanto a contenidos temáticos, desde el aspecto conjuntista contábamos con las operaciones de intersección, unión y complemento entre otras y del lado de la lógica teníamos los conectivos lógicos como las intersecciones, disyunciones y las negaciones. En un principio se pensó en desarrollar la idea de cómo los estudiantes de Ciencias de Computación asimilan las nociones conjuntistas y las nociones lógicas, es decir como diferenciaban estos dos temas matemáticos y cómo los podían articular. Los primeros obstáculos con los que nos encontramos fueron a que no contábamos con suficientes resultados sobre el área en los temas de la lógica como para poder dar comienzo a un diseño preliminar de actividades, por lo que se decidió dirigir el proyecto a aclarar cómo es que se logra el aprendizaje en un tema en particular.

Dado el enfoque de esta tesis sobre la formación de estudiantes de programación, los resultados de investigación que encontramos más útiles se enfocan en este ámbito. A continuación, presentamos los más relevantes.

1.2 Dificultades en aprendizaje de Programación relacionadas con leyes de De Morgan

Dentro del ámbito de la programación, especialmente cuando se inicia el uso de *lenguajes formales*, se encuentran algunas complicaciones en la mayoría de los lenguajes de programación como la comprensión de arreglos (listas o arrays en inglés), la recursividad, el manejo de memoria, entre otras.

Particularmente la recursividad si bien es un concepto fundamental en matemáticas y computación genera algunas dificultades tanto en programadores novatos como en aquellos que ya han generado cierta experiencia, la recursividad es una alternativa diferente para implementar estructuras de repetición (ciclos), a su vez estos se pueden implementar en cualquier situación donde la solución pueda ser expresada como una sucesión de pasos.

Uno de los primeros retos con los que se encuentra un estudiante cuando está dando los primeros pasos en la programación es el concepto de recursividad (McCauley, Grissom, Fitzgerald & Laurie Murphy, 2015, pp. 37-38). Así como los ciclos de repetición poseen clausulas condicionales que son los que controlan la repetición de este, en las funciones recursivas podemos encontrar de igual manera expresiones en las cuales se representen conectivos lógicos como disyunciones y conjunciones.

Ejemplo de la función Fibonacci recursivo:

> La función recibe como parámetro un entero; y dependerá de la condición que está dentro del cuerpo de la función el valor que esta regrese. Dentro de la condición, la función se llama a sí misma, por lo que de esta manera resulta un proceso iterativo hasta que la condición deje de cumplirse regresando el valor esperado.
>
> $Int\ factorial\ (int\ n)\{$
>
> $\quad return\ (n == 0)\ ?\ 1 :\ n * factorial(n-1);$
>
> $\}$

Varios estudios han provisto evidencia que sugiere que la actividad de recursividad es una tarea inherentemente difícil (visto en McCauley, Grissom, Fitzgerald & Laurie Murphy, 2015, p. 39).

La recursividad es uno de los temas que siempre se toman en cualquier curso básico de programación por su peculiaridad de que fomenta la parte creativa del pensamiento y de análisis de un estudiante, sin embargo, no es muy empleada actualmente, sobre todo en el mundo laboral, debido a su nivel de dificultad que requiere encontrar una solución para cada problema, y es por ello que puede ser fácilmente remplazada por ciclos de repetición.

Por otro lado, Lee y Lehrer (visto en McCauley, Grissom, Fitzgerald & Laurie Murphy, 2015, pp. 39-40) encontraron que tanto estudiantes con experiencia previa en programación, como estudiantes novatos cometen el mismo tipo de errores al realizar tareas no triviales de programación recursiva:

- Repetición innecesaria de declaraciones
- Uso inapropiado de condicionales
- Uso ineficiente de operaciones lógicas
- Inicialización inadecuada de variables.

De lo encontrado por Lee y Lehrer, suponemos que las dificultades están relacionadas con la falta de desarrollo de las leyes de De Morgan. Particularmente, una de las primeras dificultades que muestra un estudiante es el manejo de las instrucciones de control; es decir, las instrucciones de selección y las de repetición. Específicamente se presentan dificultades cuando se trabaja con las estructuras de repetición, tanto simples como anidadas (ciclos dentro de otros ciclos), y esto parece ser debido a la falta de comprensión en las condiciones que restringen tales ciclos, como lo menciona Cetin (2015).

Cetin (2015) estudió la comprensión de ciclos simples y los ciclos anidados como estructuras mentales, con un grupo de 63 estudiantes de Ingeniería Mecánica. Apoyado en el marco de la teoría APOE (acción, Proceso, Objeto, Esquema), propuso una *descomposición genética* que contempla cuatro etapas: concepción de *pre-Acción* de bucles, concepción de la *Acción* de bucles, concepción del *Proceso* de bucles, concepción *Objeto* de bucles, concepción *Objeto* de bucles de n-nivel (n bucles anidados).

Los instrumentos utilizados corresponden a dos de los que sugiere el marco teórico utilizado: el primero son los cuestionarios y el segundo las entrevistas. En el cuestionario, se plantean preguntas diseñadas para ser pertinentes de acuerdo al conocimiento que se espera que debieran tener los estudiantes, según su avance en el curso de programación. La función principal del cuestionario fue hacer visibles las concepciones de los estudiantes para clasificarlos como de bajo, medio o alto desempeño (Cetin, 2015, p. 162). El autor declara que "el propósito de las entrevistas fue producir datos ricos relacionados con la comprensión de los estudiantes sobre bucles y bucles anidados" (p. 162). Las respuestas de los estudiantes le permitieron al autor comprobar las estructuras mentales previstas y añadir una concepción no prevista.

Después de analizar los datos recabados con ambos instrumentos, se llegó a una descripción final de las concepciones de ciclos de programación, compuesta de las siguientes (Cetin, 2015, pp. 162-165):

> **Pre-Acción:**
>
> Esta etapa no se incluyó en la versión preliminar de la descomposición genética, pero los datos empíricos mostraron que es necesario añadir una nueva etapa para describir las construcciones de los estudiantes que están por debajo de la concepción de la Acción de un bucle de un nivel. Las personas limitadas a esta etapa no pueden expresar correctamente la sintaxis de un bucle. Son conscientes de que una tarea se repetirá una y otra vez, pero no pueden usar bucles para construir códigos en ejecución.
>
> **Acción**
>
> [Los individuos] pueden expresar adecuadamente la sintaxis de un bucle y pueden usarla para resolver problemas relativamente simples. Sin embargo, el concepto de bucle de un nivel tiene una naturaleza estática en esta etapa. Los individuos expresan iteraciones explícitamente de una manera paso a paso.

Proceso

> La Acción de iteración de la etapa anterior se interioriza como un Proceso. Una concepción del Proceso del bucle de un nivel tiene una esencia más dinámica. El cuerpo del bucle se realiza repetidamente hasta que la prueba indica que la iteración se detiene. Los individuos pueden imaginar iteraciones sin ejecutar realmente cada paso de la iteración.

Objeto

> Los individuos perciben el proceso de la etapa anterior como una totalidad en la que se establecen los límites de las variables de control (entrada), el bucle se ejecuta hasta que la prueba le dice que se detenga (proceso) y después se realiza el final (salida). Por lo tanto, el bucle puede considerarse como una función o procedimiento con la entrada, el proceso y la salida indicados. Esto completa la construcción del bucle de un nivel. Por lo tanto, el individuo puede construir un bucle de nivel n insertando un bucle de nivel (n - 1) en un bucle de un nivel o viceversa.

Después de analizar las concepciones mostradas por los estudiantes, Cetin (2015) plantea algunas conclusiones sobre el aprendizaje de ciclos y ciclos anidados. En particular, señala que:

> Aunque el estudiante cree que tiene un problema en los ciclos de dos niveles, el punto de vista teórico de APOE sugiere que el estudiante podría tener un problema con el entendimiento de la concepción de objeto para el bucle de un nivel porque un ciclo de dos niveles es idéntico al ciclo de un nivel, en donde al menos una de las instrucciones del cuerpo del ciclo es un ciclo de un nivel de acuerdo con la descomposición genética.
>
> Los resultados de este estudio mostraron que el bucle anidado de n niveles requiere construir la comprensión de la etapa del objeto. Hay una gran cantidad de evidencia de que los estudiantes generalmente tienen dificultades para lograr el nivel de comprensión del objeto (p.167).

El trabajo de Cetin (2015) muestra un ejemplo de la estructura esperada de una investigación realizada con APOE. Nos servirá la precisión mostrada en el análisis de estructuras mentales en contextos de programación, dado que se centró en el análisis de datos y el análisis teórico. Por otro lado, encontramos pocos trabajos dentro del rubro de la Matemática Educativa que han dirigido su atención a estas ramas de la matemática. Es por ello que partiendo de los antecedentes que se obtuvieron y de las nociones que queríamos que se mantuvieran por el lado de la lógica nos dirigimos a estudiar el concepto matemático de las leyes de De Morgan, pues este tema retoma propiedades de la lógica, algunas como: Idempotencia, Asociatividad, Distributividad, Conmutatividad, Complemento, Condicionales, Identidad, mismas que se emplean en las mismas leyes de De Morgan y que además son muy suscitadas en el ámbito de la programación sobre todo cuando se están aprendiendo los rudimentos y conceptos básicos de programación así como en niveles más avanzados donde se requiere gran habilidad de abstracción.

Para contribuir a la solución de estos problemas nos vimos en la necesidad de desarrollar una serie de actividades en las cuales se pueda modelar y desarrollar de forma secuencial la formulación de las leyes de De Morgan, representado estrictamente representado en forma de expresiones condicionales, pues es en ellas donde se presentan mayormente los casos en los problemas de programación. A su vez, estas expresiones requieren como se mencionó anteriormente el uso de operadores lógicos como la disyunción, conjunción y la negación de ambos, siendo la negación de los conectivos la que presenta una mayor dificultad. Por lo anterior, necesitamos hacer un estudio de la parte cognitiva y de la enseñanza de las leyes de De Morgan, especialmente enfocado a la intersección de la lógica proposicional y lenguajes de programación.

1.3 Inserción curricular

En la Universidad de Sonora, es común encontrar que a los estudiantes tanto del área de Informática, Computación y Electrónica se les ofrezca un curso de *Matemáticas Discretas,* el cual trata del estudio de estructuras contables a diferencia de las *Matemáticas Continuas*. Dentro del mencionado curso se estudian temas como: técnicas de conteo, lógica, arboles, grafos y algunos elementos del algebra booleana por mencionar algunos. Se incluyen tales temas en las carreras mencionadas ya que tienen aplicaciones directas en su futura práctica profesional.

En los cursos en donde hemos podido corroborar que se imparten materias tales como matemáticas discreta o cursos de lógica, fueron en carreras como ingeniería Mecatrónica, ingeniería en Electrónica, Ingeniería en sistemas de la Información, licenciatura en Ciencias de la Computación y finalmente la licenciatura de Matemáticas, particularmente nuestro interés en la carrera de Ciencias de la Computación de la universidad de sonora. En este programa los primeros acercamientos con los temas de lógica están presentes en los primeros dos semestres, comenzando con el primer semestre en la materia que tiene por nombre diseño de algoritmos la cual es donde se estudia por primera vez las leyes de De Morgan. Estas leyes se estudian solamente como una herramienta auxiliar, esto es, sin darle mucha importancia o contexto lo cual se puede comprender pues se trata de un curso introductorio. En el segundo semestre, en el curso nombrado *Matemáticas Discretas*, es cuando se le otorga un primer significado al concepto de las leyes de De Morgan y se estudia desde diferentes contextos de la computación; se comprueban igualdades empleado tablas de verdad, se simplifican circuitos lógicos y, por último, se aborda su estructura general como expresiones universales.

Anteriormente se ha señalado la existencia de una problemática en el ámbito de la lógica matemática enfocada a estudiantes que llevan cursos de programación y, en particular entre los estudiantes de la Licenciatura de Ciencias de la Computación. Identificamos que estos estudiantes están en constante uso de herramientas lógicas y generalmente tienen problemas al momento de representar expresiones lógicas. En específico, resaltamos como especialmente problemático el uso de condicionales, lo que afecta la eficiencia de los programas.

Debido a lo anterior, se ha limitado la problemática a la carrera de Ciencias de la Computación, donde se inicia el estudio de las *leyes de De Morgan*, a partir del segundo semestre. Encontramos este tema como uno de los primeros apartados del curso de *Matemáticas Discreta* y *Programación de Computadoras*, enfocándose a tareas abstractas como: comparar la equivalencia de dos expresiones usando tablas de verdad, analizar los operadores lógicos dentro de Lógica Proposicional o Álgebra Booleana y la simplificación de los circuitos lógicos. Notamos que en estos cursos solamente se dedica un breve espacio para el aprendizaje de las leyes de De Morgan; además de ser visto como una herramienta de sustitución y una serie de pasos mecánicos que posteriormente los estudiantes solo pueden aplicar si se les presentan las condiciones necesarias.

Es importante para los estudiantes que puedan encontrar patrones en contextos realistas y que puedan aplicar las leyes en tanto en códigos de programación como en el lenguaje natural; esto permitiría que desarrollen eficiencia al emplear estas leyes para optimizar el código en el lenguaje de programación, así como el tiempo de ejecución del programa.

Algunos ejemplos prácticos en donde es posible implementar estos conceptos lógicos es estrictamente en situaciones donde se presenta una expresión condicional, por ejemplo en el caso de ingresar a un sistema de seguridad es necesario ingresar alguna clave primeramente, o de igual manera una situación donde ya no se desee seguir ejecutando alguna aplicación y es necesario guardar cambios y salir, y así como este tipo de ejemplos podemos encontrar más y de mayor complejidad es decir en donde el número de cláusulas aumenten.

1.4 Objetivos de investigación

En lo expuesto anteriormente, se plantean argumentos que indican que el aprendizaje del tema matemático de las leyes de De Morgan no es trivial. Si este se reduce a emplearlo de forma mecánica y a la simple memorización, se ocasionan dificultades sobre todo en los estudiantes que trabajan con lenguajes de programación pues constantemente están trabajando con expresiones lógicas.

Con base en lo anterior, nos proponemos realizar un estudio desde el punto de vista cognitivo del tema de las leyes de De Morgan, Nos interesa saber cómo se construye este tema a partir de los conectivos lógicos, además de ver qué tipo de construcciones mentales son necesarias para su aprendizaje. Por lo tanto, nos planteamos como **Objetivo General:**

> Aclarar cómo podrían aprender el concepto matemático de las leyes de De Morgan estudiantes de Ciencias de Computación, definiendo construcciones y mecanismos mentales asociados a la construcción del concepto.

Para alcanzar este objetivo, se realiza un estudio basado en el ciclo de investigación de APOE en el que se analiza el concepto de las leyes de De Morgan y cómo se aplica en distintos contextos en computación. De acuerdo con las recomendaciones del marco de APOE, para avanzar en el cumplimiento del objetivo general nos planteamos también los siguientes **Objetivos Específicos:**

1. Diseñar instrumentos para obtener información sobre el aprendizaje de las leyes de De Morgan, en particular una serie de actividades que permitan evaluar las construcciones mentales que utilizan los estudiantes al resolver problemas.
2. Desarrollar una descomposición genética preliminar, que describa un panorama cognitivo y que pueda fundamentarse para propósitos posteriores a un diseño de actividades que promuevan las tareas de reflexión y aprendizaje del concepto matemático.
3. Proponer recomendaciones para la enseñanza del tema a partir de los resultados obtenidos. Por ejemplo, describir los tipos de actividades o tareas necesarios para la construcción del concepto por los estudiantes.

1.5 Preguntas de investigación

Así mismo, para contribuir a una solución a la problemática descrita, nos planteamos responder en este trabajo a la siguiente pregunta principal:

- ¿Cómo construyen el concepto matemático de las Leyes de De Morgan los estudiantes de carreras que llevan cursos de programación?

Para desarrollar la respuesta a la pregunta anterior, nos planteamos también responder:

- ¿Qué construcciones previas son necesarias para desarrollar el concepto de las leyes de De Morgan?
- ¿Cómo se caracteriza la transición de una concepción a otra en el desarrollo de este concepto?
- ¿Qué tipo de estrategias pedagógicas le pueden ayudar al estudiante para la construcción de este concepto matemático?

2 Marco Teórico

Nuestro trabajo está apoyado en el marco teórico constructivista APOE. Sus siglas significan Acción, Proceso, Objeto, Esquema que son las estructuras mentales que describen la manera en que se construyen los conceptos matemáticos dentro de la abstracción reflexiva. Salgado y Trigueros (2009) describen la teoría de la siguiente manera:

> La teoría APOE proporciona una base teórica para analizar la forma en la que se construyen los conceptos matemáticos para estudiar cómo evolucionan las construcciones de los alumnos y, al mismo tiempo, permite diseñar estrategias didácticas para ayudar a los alumnos a hacer las construcciones necesarias para que esta evolución tenga lugar y se logre un aprendizaje de los conceptos matemáticos más significativos. Los componentes esenciales de la teoría APOE son: acciones, procesos, objetos y esquemas, además de la abstracción reflexiva que se considera como mecanismo de construcción; este mecanismo da lugar a los procesos deconstrucción de conocimiento, como son la interiorización, la generalización, la coordinación, la encapsulación y la reversión (p. 94).

2.1 La abstracción reflexiva

La visión más importante de Piaget fue la de idear el concepto de la abstracción reflexiva la cual básicamente se puede representar como que “el sujeto extrae información de sus propias acciones sobre los objetos”. La forma en que Piaget lo percibía fue de dos maneras, la primera tenía que ver con todo lo que implicaba la reflexión y el pensamiento. La segunda manera consistía en que el conocimiento se proyectaba sobre un plano superior en el cual este se organizaba y se reconstruía para obtener como resultado un nuevo conocimiento, esta segunda manera la retomó Dubinsky pues le pareció muy cercano a ciertas ideas que se contemplaban en matemáticas y con ello le permitió creer que la abstracción reflexiva podría ser una gran herramienta para permitir el desarrollo mental en conceptos avanzados en matemáticas.

Un ejemplo de cómo se podría interpretar la abstracción reflexiva dentro del tema de las leyes de De Morgan, tomando como referencia el segundo paso basado en el de Piaget, seria el caso en que el individuo conoce en su conjunto los conectivos lógicos. Si se emplean operaciones entre estos conectivos, surgirán como resultado nuevas operaciones y expresiones que llamaremos reglas o tautologías.

El fundamento principal de la teoría APOE es la denominada abstracción reflexiva, pues es por medio de esta que las construcciones mentales se consiguen. La abstracción reflexiva se compone de cinco mecanismos los cuales son *interiorización, coordinación, encapsulación, generalización y reversión.* Estos mecanismos son lo que permiten pasar de una concepción mental a otra. Un ejemplo de ello sería de pasar de la concepción Acción a Proceso, esto es posible solamente cuando existe internamente una interiorización.

2.2 Las construcciones mentales

Como ya fue mencionado las construcciones mentales son Acción, Proceso, Objeto y Esquema. Estas integran lo que denominaremos etapas, las cuales forman la construcción del conocimiento; cabe resaltar que dichas etapas no son necesariamente secuenciales y tampoco siguen un orden específico. A continuación, proporcionaremos una descripción de cada etapa.

2.2.1 Acción

De acuerdo al marco teórico una Acción es un estímulo externo que experimenta el individuo sobre objetos previamente concebidos, conocidas como las construcciones previas. Una acción contempla una serie de transformaciones las cuales se le conocen como pasos. Estos pasos son los elementos de entrada, las operaciones entre estos y los elementos de salida. Para que una acción pueda llevarse a cabo es necesario que los pasos mencionados se lleven a cabo sin la intervención de ningún salto entre ellos y que además no puedan ser imaginados por los individuos; para poder lograrlo, estos se presentan de forma explícita y a su vez guiada por instrucciones secuenciales en que cada paso promueva al siguiente.

2.2.2 Proceso

Cuando una acción se repite constantemente el individuo empieza a tomar conciencia sobre lo que está efectuando; esto lo lleva a interiorizar la acción en proceso. El proceso se logra construir con los mecanismos de interiorización o coordinación. Decimos que el proceso desempeña la misma operación que la acción solo que esta es interiorizada y además se ejecuta en la mente del individuo sin necesidad de ejecutar los pasos requeridos en la etapa de acción.

2.2.3 Objeto

En esta concepción el individuo ya puede efectuar operaciones sobre los Procesos, lo cual resultaría en la obtención de nuevo procesos. La manera de identificar que el individuo está empleando la concepción Objeto es que el individuo usa el Proceso como un todo y puede emplear ciertas transformaciones como lo son las Acciones y Procesos. En este caso podemos decir que el individuo ha alcanzado cierta madurez, por lo que ha adquirido la habilidad de identificar en qué momento está efectuando las construcciones de Acción o Proceso. Los mecanismos que están presentes en esta construcción son los de desencapsulación y encapsulación; esto significa que el individuo es capaz de moverse de Objeto a Proceso con el fin de corroborar de dónde obtuvo sus resultados.

2.2.4 *Esquema*

Un esquema se puede interpretar como el conjunto de acciones, procesos, objetos y otros esquemas que a su vez son usados para dar soluciones a problemas en matemáticas,

Al esquema se le puede considerar como un nuevo objeto; es decir, de la misma manera cuando el proceso avanza y se coordina con otros procesos se logra concretar en un objeto, el esquema se logra cuando el objeto se tematiza y es considerado como un mecanismo que permite al individuo resolver un problema.

2.3 Descomposición Genética

Una descomposición genética es el trabajo que culmina tras haber implementado el ciclo de investigación y que particularmente surge en la etapa del análisis teórico por lo tanto una descomposición genética es un modelo hipotético y preliminar en el cual están involucrados las estructuras mentales y los mecanismos que un individuo puede necesitar en la construcción y aprendizaje de un concepto matemático particular a partir de las construcciones previas. Además, la descomposición genética se utiliza como base teórica para diseñar materiales de enseñanza los cuales son llevados al salón de clase para posteriormente seguir haciendo análisis de los resultados.

3 Metodología

Una característica notable del marco APOE es que incluye una metodología de investigación: el paradigma o ciclo de investigación de APOE (Arnon et al., 2014, p. 96.) se muestra en la Figura 4.1. “Este ciclo procura conseguir una descripción próxima de la construcción de los conceptos matemáticos y tener una mirada más cercana y detallada del proceso de construcción por parte de los estudiantes en relación a los conceptos que se desean investigar” (Gamboa, 2013, p. 30).

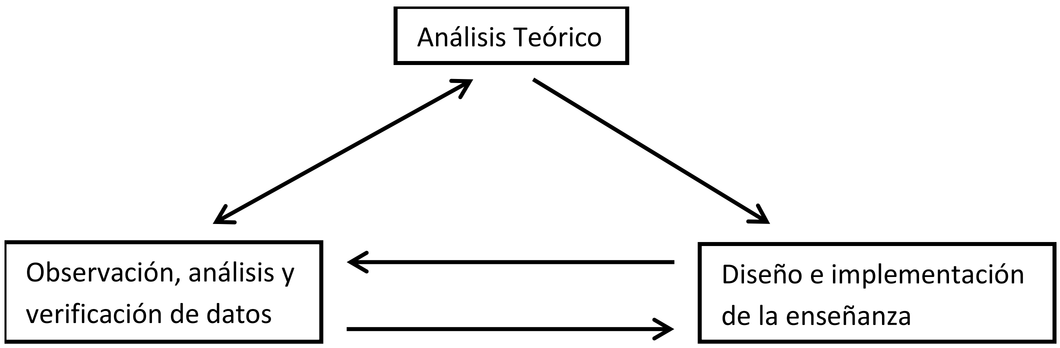


Fig. 3.1 Ciclo de investigación APOE

Los elementos del ciclo APOE son el análisis teórico, diseño e implementación de la enseñanza y la observación y verificación de datos. Cada etapa genera un producto específico:

3.1 Análisis teórico

Esta etapa surge de un análisis que se efectúa a partir del análisis del concepto matemático de estudio. Para lograrlo es necesario hacer una investigación previa la cual puede surgir a partir de libros de texto y revistas de divulgación científica. Todo esto se realiza con el fin de obtener un modelo cognitivo que permita definir una descripción viable de las construcciones mentales del concepto matemático. Además, uno de los objetivos más sobresalientes del análisis teórico es que esta etapa produce la Descomposición Genética (preliminar).

3.2 Diseño e implementación de la enseñanza:

Uno de los principales propósitos de esta etapa es corroborar la descomposición genética preliminar que se propuso en la etapa del análisis teórico; para ello se echa mano de algunos instrumentos de evaluación los cuales pueden ser reactivos como preguntas abiertas y ejercicios donde se necesite realizar cálculos, así como el empleo de medios tecnológicos para su realización. Estos instrumentos son diseñados con el propósito de identificar las construcciones mentales que se previeron en la descomposición genética.

3.3 Observación, Análisis y verificación de los datos:

En esta etapa se lleva a cabo la recolección de información de la etapa posterior de enseñanza. El objetivo de esta etapa es la evaluación de la pertinencia de la Descomposición Genética preliminar y la secuencia de actividades de enseñanza. En esta etapa se analizan tanto elementos y construcciones mentales esperadas, como todo aquel elemento que no fue considerado en la etapa del análisis teórico, así de esta manera esta etapa conlleva a una reestructuración de la descomposición genética más depurada.

Habiendo mencionado el ciclo de investigación de APOE y sus características más notorias, guiados en este, presentaremos a continuación el ciclo que se siguió en este trabajo (Figura 4.2). Cabe mencionar que una de las características con las que cuenta el marco de investigación que estamos empleando, es que nos brinda la flexibilidad de poder implementar ajustes a nuestras necesidades particulares; por lo tanto, para responder a las preguntas de investigación se implementaron seis etapas que explicaremos a continuación.

Si bien en el ciclo de investigación mostrado es posible comenzar en cualquiera de las tres etapas dependiendo del fin que se desee realizar, para nuestro trabajo fue necesario partir del análisis teórico y la primera etapa, debido a que no contábamos con resultados previos al tema matemático de las leyes de De Morgan. De esta manera caracterizamos algunas de las concepciones metales, las cuales consideramos que serían las que más favorecerían el aprendizaje de los estudiantes en que estábamos enfocados. Nos centramos en las Acciones y los Procesos. Posteriormente, pasamos a la etapa del diseño y la implementación de la enseñanza que en nuestro trabajo particular fueron actividades enfocadas a la resolución de ejercicios en papel, así como la intervención de medios tecnológicos. Las actividades propuestas tienen un enfoque de exploración en contextos reales diversos; su estructura es semejante a un examen, puesto que lo que pretendíamos era evaluar las concepciones mentales relacionadas con el tema matemático. Tras la puesta en práctica de las actividades, en la tercera etapa pudimos definir concepciones específicas sobre cada ley y su coordinación apoyándonos en el análisis de los resultados y el análisis a priori. Para la cuarta etapa fue necesario comprobar las concepciones mentales que definimos previamente en el análisis teórico de las respuestas basadas en el marco con los resultados de los participantes. Por medio de los resultados proporcionados por los análisis anteriores, en la quinta etapa contamos con el sustento necesario para poder proporcionar una descomposición genética preliminar; esta nos brinda información sobre cómo es que se podría lograr el aprendizaje y permite diseñar la enseñanza e instrumentos que favorezcan las concepciones mentales buscadas, es decir, permite dirigirse nuevamente al diseño de la enseñanza, en nuestro caso, por cuestiones de tiempo, nos limitaremos a dar recomendaciones de enseñanza.

Con respecto a la descomposición genética la llamaremos preliminar hasta que pueda ser evaluada en iteraciones posteriores en el ciclo APOE.

En resumen, para el desarrollo del proyecto se siguieron las siguientes etapas:

1. Análisis teórico
   1.1. Analizar resultados de investigación sobre el tema relacionados con las leyes de DM
   1.2. Caracterizar concepciones Acción
   1.3. Caracterizar concepciones Proceso

      Estas caracterizaciones no pretenden ser sólidas desde el punto de vista del marco teórico, puesto que en ese momento no contábamos con definiciones concretas de estas concepciones; sin embargo, sí se encontraron algunas características generales de las concepciones tales como dificultades en los operadores disyunción y conjunción con las cuales pudieron guiarnos.
   1.4. Permite iniciar el diseño
2. Diseño e implementación de actividades de exploración
   2.1. Actividades que permitan evaluar las concepciones mentales usadas al resolver problemas de las leyes de De Morgan
   2.2. Implementación con estudiantes de Ciencias de la Computación
   2.3. Permite la recolección de datos
3. Definir concepciones mentales sobre las leyes, a partir del análisis de algunos de los problemas propuestos
4. Comprobar la presencia de las concepciones definidas en el grupo de participantes
5. Generar una Descomposición Genética preliminar
6. Dar recomendaciones de enseñanza a partir de la Descomposición genética preliminar.

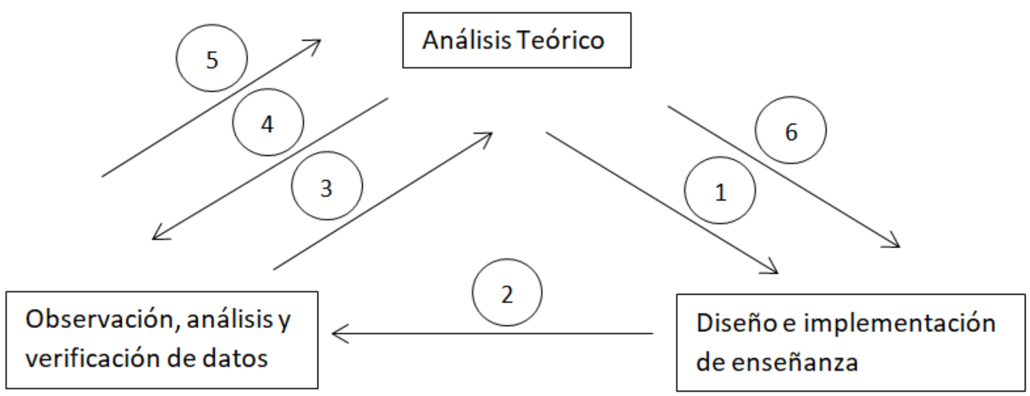


Fig. 3.2 Ciclo Implementado

4 Diseño y análisis a priori de actividades exploratorias

4.1 Diseño de problemas y respuestas esperadas

Antes de comenzar esta sección es necesario hacer algunas aclaraciones sobre el enfoque de las actividades propuestas, pues estas actividades como lo hemos mencionado anteriormente no son de carácter de enseñanza sino de exploración. Ahora bien, que sean actividades de exploración significa que de estas obtendremos solamente datos útiles para poder dar visibilidad a las construcciones mentales del concepto matemático, con el fin de lograr una descomposición genética preliminar.

Estas actividades de exploración están enfocadas a concepciones Acción y Proceso, a partir del estudio de una serie de investigaciones previas (Cetin, 2015; Vest, 1981; Macbeth, Razumiejczyk & Campitelli, 2013), en los que, si bien nuestro tema no fue directamente analizado, si se presentaba de manera implícita. Recurrimos a libros de texto que presentan las definiciones clásicas de las leyes de De Morgan y algunos ejercicios, se buscó en foros de internet donde se trataban temas como las dificultades y ventajas del uso de las leyes en contextos de programación además se apoyó también en el material que se usa para impartir los cursos tanto en *Matemáticas Discretas* como los de *Programación de Computadoras* que son correspondientes a los primeros dos semestres de la carrera de *Ciencias de la Computación* y, una vez teniendo algunos referentes, fue necesario hacer la intervención bajo nuestro propio criterio para el diseño de las actividades. Para estas actividades se decidió también emplear los medios tecnológicos para algunos ejercicios, la razón fue fomentar los medios tecnológicos en ambientes de enseñanza propicia a que se tenga una mayor comprensión del objeto matemático. Es decir, este enfoque ayuda al estudiante a orientarse más en la comprensión del concepto matemático en lugar de sus operaciones, esto además propicia la exploración del enfoque de las construcciones metales en colaboración con estos medios.

De tal manera, cuando en la descripción de las actividades se menciona "promover concepciones mentales", nos referimos a promover que los estudiantes utilicen las concepciones con herramientas que ya cuenten a su disposición, para poder observarlas y posteriormente analizarlas y definirlas.

A continuación, se presenta el diseño de las actividades desde el marco teórico APOE. Las construcciones se promueven en una serie de tres actividades principales y una actividad preliminar para evaluar si el individuo posee las construcciones previas necesarias para continuar con las actividades.

Con el objetivo de dar contexto y claridad a esta sección las actividades se presentarán el orden siguiente: primero se mostrarán las respuestas que se espera que los estudiantes puedan generar: posteriormente se proporcionará el análisis correspondiente a los elementos teóricos.

4.1.1 Actividad 0

Sección 1.

Para esta actividad introductoria, se pretende ver si lo estudiantes poseen ciertas construcciones previas las cuales nos dará un indicativo si es que pueden seguir avanzando en las actividades posteriores.

En el ambiente dinámico para esta actividad (Figura 5.1), se presenta recta numérica, en la cual mediante un botón se van generando desigualdades mismas que hay que representar en la recta.

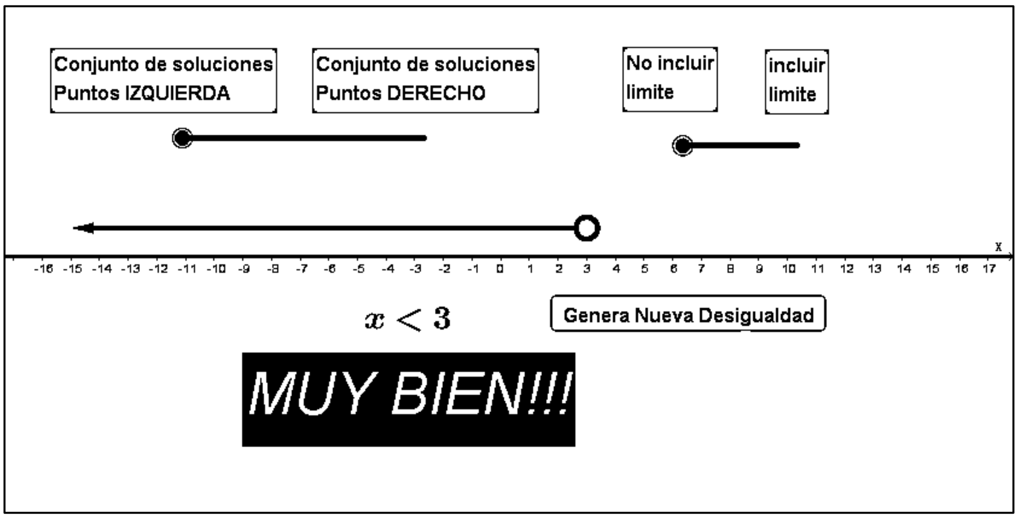


Fig. 4.1

Se espera que los estudiantes generen cuatro expresiones aleatorias mismas que escribirán en la tabla (Figura 5.2), seguidos representar su expresión negada como se muestra en la siguiente tabla, esto nos proporcionará información si es que los individuos cuentan con nociones de lo que llamamos construcciones previas.

Desigualdades generadas	Su valor negado
$x < 1$	$x >= 1$
$4 > x$	$4 <= x$
$7 > x$	$7 <= x$
$-9 <= x$	$-9 > x$

Fig. 4.2

Sección 2.

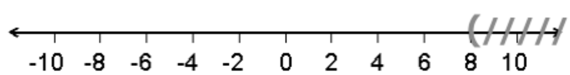
i. En la desigualdad $\boldsymbol{x \leq 8}$, qué valores está tomando la $\boldsymbol{x}$, según los siguientes incisos, marca la desigualdad más conveniente y escribe tus observaciones y porque elegiste esa opción.

$\boldsymbol{x < 8 \text{ or } x = 8}$ $\qquad\qquad$ $\boldsymbol{x < 8 \text{ and } x = 8}$

Debido a que la expresión es verdadera cuando alguno de los dos valores es asignado, se espera que el individuo selecciónela expresión $\boldsymbol{x < 8 \text{ or } x = 8}$, y que en sus argumentos pueda diferenciar que la expresión $\boldsymbol{x \leq 8}$ es condicionada y que solo puede tomar uno de los dos valores y no ambos.

ii. Ahora si tomamos a $\boldsymbol{P}$: $\boldsymbol{x \leq 8}$ como expresión, qué región obtendríamos con la expresión $\boldsymbol{not(P)}$, represéntalo en la gráfica.

Se espera que debido a la experiencia generada en los ejercicios anteriores en donde estuvo trabajando con el ambiente dinámico, el individuo pueda ser capaz de sombrear sobre la recta, pues esta representa la negación de la expresión dada.



iii. En la desigualdad $\boldsymbol{-5 < x}$, qué valores tendríamos en la gráfica si tomamos todos los que no pertenecen a ese intervalo, señálalo en la gráfica y argumenta tu respuesta.

Esta pregunta es similar a la anterior, solo que esta ocasión la negación esta expresada de forma literal y no simbólica, por lo que se espera el sombreado como se muestra en recta (Figura 5.3) represente la negación de la expresión sobre la recta, dentro de los argumentos se pueden considerar respuestas tales como, “es su valor contrario”, “la parte sombreada representa su negación”, entre otras posibles respuestas**.**

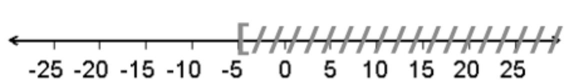


Fig. 4.3

Sección 3.

i. Cuál es la salida de los siguientes códigos

```
//CODIGO 1
#include <stdio.h>
#include <iostream>
using namespace std;
int main(){
    int x=0;
    for(int i = 1; i < 15 ; i++) {
         x=2*i;
        printf("\n%d",x);
           }
       }
```

Código 1: Es probable que el individuo marque la salida del código paso por paso, es decir: 2,4,6,8,10,12,14,15,18,20,22,24,26,28. O que diga directamente que la salida dará como resultado los primeros 14 números pares.

Código 2: se espera que lleguen a que la suma de los primeros 9 números da como resultado 45.

Sección 4.

i. A continuación escribe el resultado que se obtiene al emplear el operador lógico negación, sobre cada expresión.

$not(x < 6) =$

$not(not(x \geq 9)) \ =$

Se espera que los estudiantes puedan emplear de manera correcta el operado lógico negación y poder proporcionar el siguiente resultado:

$$not(x < 6) = x \geq 6$$

$$not\big(not(x \geq 9)\big) = x \geq 9$$

4.1.2 Actividad 1

Sección 1.

En esta sección se pretende que los individuos puedan manejar adecuadamente la información que se les proporciona para de alguna manera ir ayudando y provocando estímulos a los conceptos que pretendemos trabajar en las próximas secciones y actividades, dentro de toda la actividad uno, pretendemos evidenciar la concepción Acción como lo estipula el marco teórico, la cual se le propone al individuo ejercicios, con los cuales pueda empezar a operar sobre ellos generando a su vez salidas, con las cuales pueda construir reflexiones.

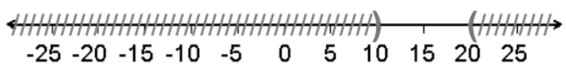
Se estarán trabajando recurrentemente las desigualdades, debido a que son muy empleadas al momento de modelar alguna condición o comportamiento dentro del ámbito de programación.

a) Primer caso

i. Representa las siguientes expresiones en su grafica correspondiente.

Se espera que los estudiantes puedan manejar la notación de los intervalos abiertos y cerrados, además que puedan hacer el manejo debido de los operadores lógicos (not, and, or), por lo que el resultado esperado se muestra en las siguientes graficas (Figura 5.4).

$$(\boldsymbol{x} < \mathbf{10}\ \boldsymbol{or}\ \boldsymbol{x} > \mathbf{20})$$



$$\boldsymbol{not}(\boldsymbol{x} >= \mathbf{10}\ \ \boldsymbol{and}\ \ \boldsymbol{x} <= \mathbf{20})$$

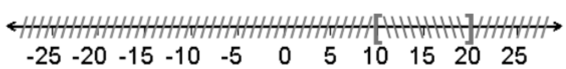


Fig. 4.4

ii. Compara ambas gráficas y escribe que relación encontraste en ellas.

Una vez concluido el trabajo con las gráficas, como argumentos importantes serían los siguientes: podrían decir que en ambas graficas se están representando los mismos valores y que eso es posible debido al operador negación que interviene en la segunda expresión es por lo tanto que argumenta que ambas graficas representan valores iguales.

iii. Llenado de la tabla para $(\boldsymbol{P} \wedge \boldsymbol{Q})$.

Se espera que el individuo ingrese el valor que le fue generado al ingresar la desigualdad en el ambiente dinámico sobre las columnas de $(\boldsymbol{P} \wedge \boldsymbol{Q})$ y pueda encontrar a su vez su correspondiente negación, así obteniendo la siguiente salida.

Caso	$\boldsymbol{P}$	$\boldsymbol{Q}$	$\boldsymbol{R}$	$\boldsymbol{not(R)}$
1	$\boldsymbol{x} > \mathbf{10}$	$\boldsymbol{x} \leq 20$	$(\boldsymbol{x} > \mathbf{10})\ \boldsymbol{and}\ (\boldsymbol{x} <= \mathbf{20})$	$\boldsymbol{not}(\boldsymbol{x} > \mathbf{10})\ \boldsymbol{or}\ \boldsymbol{not}(\boldsymbol{x} <= \mathbf{20})$
2	$\boldsymbol{x} > 5$	$\boldsymbol{x} \leq 9$	$(\boldsymbol{x} > \mathbf{5})\ \boldsymbol{and}\ (\boldsymbol{x} <= \mathbf{9})$	$\boldsymbol{not}(\boldsymbol{x} > \mathbf{5})\ \boldsymbol{or}\ \boldsymbol{not}(\boldsymbol{x} <= \mathbf{9})$
3	$\boldsymbol{x} > -3$	$\boldsymbol{x} \leq 12$	$(\boldsymbol{x} > \mathbf{-3})\ \boldsymbol{and}\ (\boldsymbol{x} <= \mathbf{12})$	$\boldsymbol{not}(\boldsymbol{x} > \mathbf{-3})\ \boldsymbol{or}\ \boldsymbol{not}(\boldsymbol{x} <= \mathbf{12})$

4	$\boldsymbol{x} \geq 0$	$\boldsymbol{x} \leq 9$	$(\boldsymbol{x} >= \mathbf{0})$ $\boldsymbol{and}$ $(\boldsymbol{x} <= \mathbf{9})$	$\boldsymbol{not}(\boldsymbol{x} >= \mathbf{0})$ $\boldsymbol{or}$ $\boldsymbol{not}(\boldsymbol{x} <= \mathbf{9})$

iv. ¿Qué observaste en las gráficas que obtuviste al ingresar los valores?, ¿tuviste alguna complicación?

Una vez completado el llenado de la tabla y el haber observado las representaciones graficas que fueron mostradas por el applet, es deseable que los individuos generen argumentos como: “la expresión de la derecha siempre es la respuesta esperada pues siempre es su valor contrario”, o “las gráficas representan el mismo valor cuando estas dos expresiones son seleccionadas”.

v. ¿El applet calculó bien los valores de R?, ¿Qué usaste para decidir tu respuesta?

En el mejor de los casos se espera que los individuos no hayan generado alguna complicación en el manejo del applet, por lo que se espera que respondan de manera afirmativa, para la segunda pregunta es posible que la respuesta la hayan generado mediante otros medios ya sean lápiz y papel o cálculos mentales, sin embargo, es deseable que su respuesta sea basada totalmente en la herramienta manipulable.

vi. ¿Cómo escribirías en términos de $\boldsymbol{P}$ y $\boldsymbol{Q}$ las expresiones del lado derecho del applet de $\boldsymbol{R}$ y $\boldsymbol{not}(\boldsymbol{R})$?

Se espera como resultado la siguiente tabla (Figura 5.5) y si por alguna razón no pudieron con la tabla anterior se espera que al menos sí puedan responder esta tabla.

caso	$\boldsymbol{R}$	$\boldsymbol{not}(\boldsymbol{R})$
1	$\boldsymbol{P\ and\ Q}$	$\boldsymbol{notP\ or\ notQ}$
2	$\boldsymbol{P\ and\ Q}$	$\boldsymbol{notP\ or\ notQ}$
3	$\boldsymbol{P\ and\ Q}$	$\boldsymbol{notP\ or\ notQ}$
4	$\boldsymbol{P\ and\ Q}$	$\boldsymbol{notP\ or\ notQ}$

Fig. 4.5

vii. ¿Hay algo que siempre se cumpla al ingresar los valores en la tabla?

Se espera que respondan que la negación de una and es equivalente al negado de los or.

viii. ¿La expresión $\boldsymbol{not}(\boldsymbol{P} \wedge \boldsymbol{Q}) = \boldsymbol{notP} \wedge \boldsymbol{notQ}$ es posible el poder emplearla?, pruébalo en el applet y después escribe tus observaciones

Hasta este momento se han ido presentando diversos contextos como lo fueron las desigualdades, expresar las expresiones sobre la recta y el llenado de tablas, para esta pregunta su propósito nos ayuda a dirigir la secuencia de la actividad, pues por una parte exigen concentración del individuo a que formule una respuesta objetiva, y por otro lado nos proporciona detalles si es que se está siguiendo la secuencia o de otra forma se debería de reformular las actividades.

La respuesta a esta pregunta es evidente y los argumentos que serían favorables son: “no es posible emplearla, pues no hay valores al ingresar el en applet que satisfagan la condición”.

b) Segundo caso

i. Representa las siguientes expresiones en su grafica correspondiente.

De igual manera, así como en el primer caso se espera que los estudiantes puedan manejar la notación de los intervalos abiertos y cerrados, además que puedan hacer manejo debido de los operadores lógicos *(not, and, or),* el resultado óptimo es que puedan sombrear las secciones y a partir de ellas poder generar una solución correcta como se muestra en las siguientes rectas (Figura 5.6).

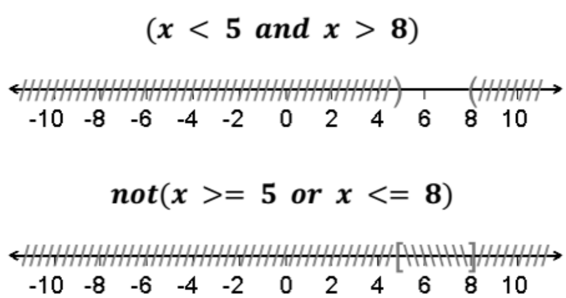


Fig. 4.6

ii. Compara ambos resultados de las gráficas y determina qué relación encontraste entre ambos casos, anota tus observaciones.

De igual manera a como se expuso en el primer caso, se esperan argumentos referentes a la similitud de ambas rectas; sin embargo al igual que en el primer caso, podría causar algunas dificultades, pues se podría interpretar que la negación del conjunto solución $not(x >= 5 \ or \ x <= 8)$ es que no corresponde a ningún valor dentro del intervalo, pasando por alto que la solución es el complemento de ese conjunto.

Accede al siguiente, así como lo hiciste en el inciso a) del ejercicio anterior link: **<u>http://www.mat.uson.mx/~teoriadeconjuntos/</u>**

iii. llenado de la tabla para $(\boldsymbol{P} \vee \boldsymbol{Q})$.

A partir de la tabla que se proporciona en las hojas de trabajo (Figura 5.7) se espera que el individuo ingrese el valor que le fue generado al ingresar la desigualdad en el ambiente dinámico sobre las columnas de $(\boldsymbol{P} \vee \boldsymbol{Q})$ y pueda encontrar a su vez su correspondiente negación, así obteniendo la siguiente salida.

Caso	P	Q	R	$not(R)$
1	$x < 10$	$x > 20$	$(x < 10) or (x > 20)$	$not(x < 10)$ and $not(x > 20)$
2	$x < 5$	$x > 9$	$(x < 5) or (x > 9)$	$not(x < 5)$ and $not(x > 9)$
3	$x < -3$	$x > 12$	$(x < -3) or (x > 12)$	$not(x < -3)$ and $not(x > 12)$
4	$x < 0$	$x > 9$	$(x < 0) or (x > 9)$	$not(x < 0)$ and $not(x > 9)$

Fig. 4.7

iv. ¿El applet calculo bien los valores de R?, ¿Qué usaste para decidir tu respuesta?

El resultado esperado es que pueda responder un Si, y en sus argumentaciones se espera que contesten: "porque siempre que ingreso el valor a encontrar siempre me muestra que una de las opciones siempre es correcta."

También se puede considerar que el individuo opte por el uso de papel y represente los valores sobre una línea recta para corroborar su respuesta.

A continuación, se muestra con detalle cómo es que se interactúa con el applet para el llenado de la tabla para el ejercicio iv.

Cada expresión de los valores P y Q de la tabla, se introduce en los espacios de P y Q del manipulable (en la parte superior izquierda de la figura 5.8) y se asigna el caso que se está trabajando, de una lista desplegable, en este caso es el operador conjunción.

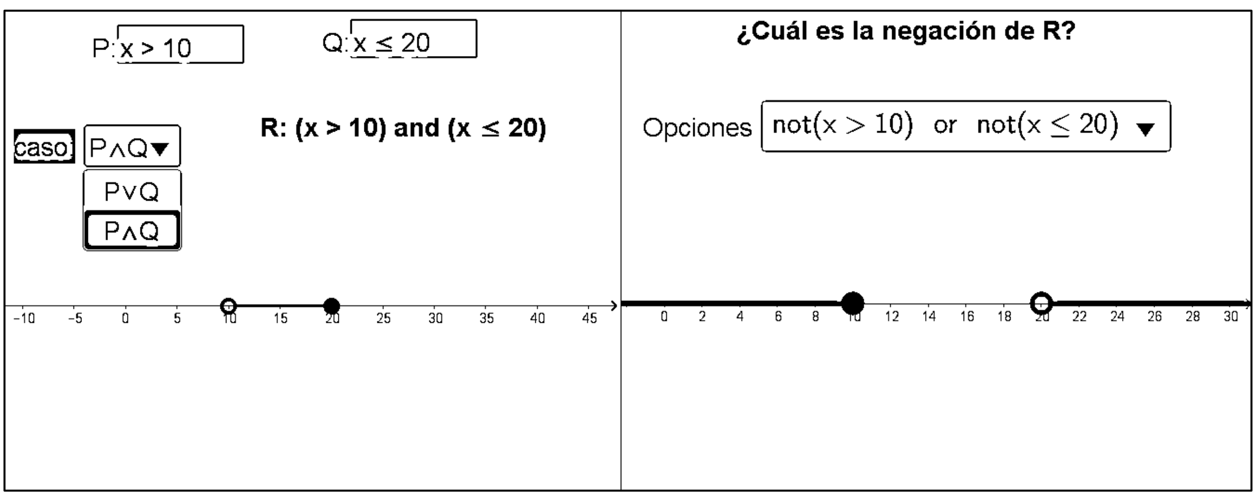


Fig. 4.8

Del lado derecho del applet (Figura 5.9), se pide que se encuentre la correcta expresión de la negación de la proposición anterior, seleccionando dentro de la lista de opciones, como se muestra en la siguiente figura.

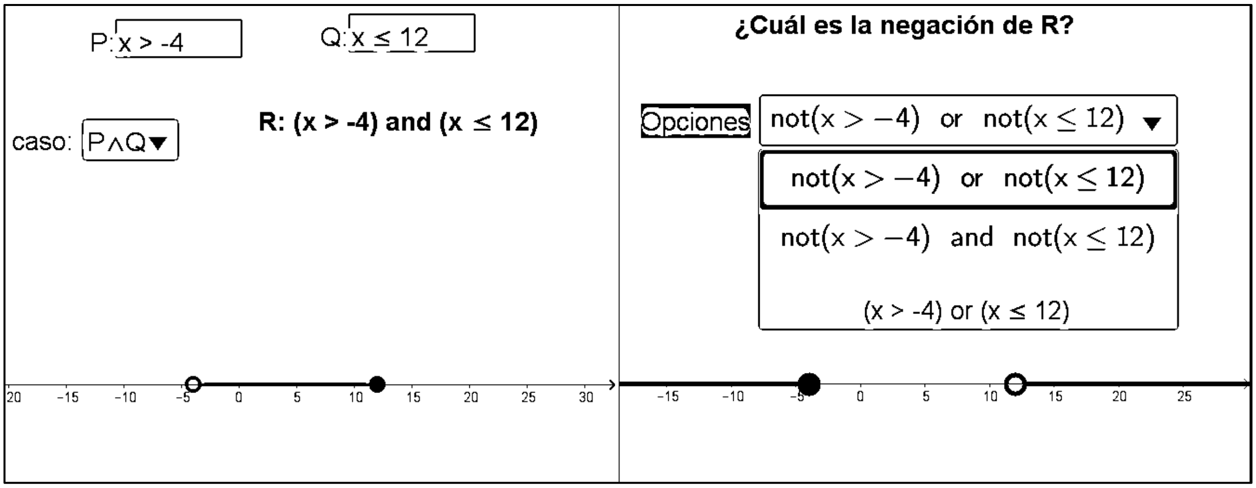


Fig. 4.9

v. ¿Cómo escribirías en términos de $\boldsymbol{P}$ y $\boldsymbol{Q}$ las expresiones de $\boldsymbol{R}$ y $\boldsymbol{not(R)}$?

Basta con que el individuo llene los campos, tal y como es mostrado en la tabla de abajo (Figura 5.10).

caso	$\boldsymbol{R}$	$\boldsymbol{not(R)}$
1	$\boldsymbol{P\ or\ Q}$	$\boldsymbol{notP\ and\ notQ}$
2	$\boldsymbol{P\ or\ Q}$	$\boldsymbol{notP\ and\ notQ}$
3	$\boldsymbol{P\ or\ Q}$	$\boldsymbol{notP\ and\ notQ}$
4	$\boldsymbol{P\ or\ Q}$	$\boldsymbol{notP\ and\ notQ}$

Fig. 4.10

vi. ¿Hay algo que siempre se cumpla al ingresar los valores en la tabla?

Con este tipo de preguntas se espera que el individuo logre la reflexión y aunado con la experiencia que ha adquirido hasta este momento se espera que puedan argumentar que la negación de una or es equivalente al negado de los and.

vii. ¿La expresión $\boldsymbol{not(P \vee Q) = notP \vee notQ}$ es posible el poder emplearla?, pruébalo en el applet y después escribe tus observaciones

Se espera un No como respuesta y como argumento mencionen que no es posible emplearla pues no hay valores al ingresar en el applet que satisfagan la igualdad.

Sección 2.

En esta sección, lo que se pretende es poner al estudiante en un contexto externo y sobre otro ambiente, donde tiene que manejar los mismos conceptos que ha estado trabajando anteriormente, para ello utilizamos el juego de ajedrez y un ambiente de programación en C++ del cual se espera que ya estén familiarizados.

a) primer caso

i. Ejecuta el código A y B, usa el diagrama para ayudarte y ve marcando sobre él, las iteraciones que creas conveniente.

Se espera que los individuos interpreten el código A y B y puedan representar la salida en el diagrama (Figura 5.11) como se les indicó obteniendo la salida que está representada en el diagrama, además del diagrama existe la posibilidad de algunas cuentas mentales y escritos fuera del diagrama.

	0	1	2	3	4	5	6	7
0	✓							
1	X							
2	X							
3	X							
4								
5								
6								
7								

Fig. 4.11

ii. ¿Qué es lo que están haciendo los códigos A y B? explique su salida.

Se espera que argumenten que en el código son los movimientos que están realizando las piezas blancas del diagrama.

iii. ¿Crees que es conveniente para la maquina el haber optado por utilizar alguno de los dos códigos?, ¿Por qué?

Posiblemente el individuo escogerá el código que le resulto más fácil de leer a nivel de las expresiones lógicas a pesar de ser equivalentes, el hecho está en que la pregunta pide escoger uno de los dos.

Se espera que argumenten que ambos códigos producen salidas iguales.

iv. Para el código B, ¿si quitamos la condición $!(\boldsymbol{Peon}[\boldsymbol{i}-\mathbf{1}][\mathbf{0}]\,!=\boldsymbol{NULL}$ dentro del ciclo $\boldsymbol{while}$, consideras que se pueda seguir ejecutando el juego para ese caso particular? ¿Por qué?

Se espera que mencione que el juego se sigue ejecutando pues en el operado lógico or solo le es suficiente que una de las dos expresiones se cumpla.

Tal vez no se pretende que en este punto logren una reflexión compleja de dentro de las leyes de De Morgan; a pesar de ser equivalentes, estas pueden afectar a las expresiones en casos particulares como este.

b) Segundo caso

i. ¿Cuál es el resultado que obtuviste con los códigos C y D?

Se espera que los individuos analicen el código C y D y puedan representar la salida en el diagrama (Figura 5.12) como se les indico obteniendo la salida que está representada en el diagrama.

	0	1	2	3	4	5	6	7
0								
1								
2								
3		x						
4		x						
5		x						
6		x						
7		✓						

Fig. 4.12

i. ¿Cuál es el resultado que obtuviste con los códigos C y D?

Se espera que argumenten que en el código son los movimientos que están realizando las piezas negras del diagrama.

ii. ¿Crees que es conveniente para la maquina el haber preferido utilizar alguno de los dos códigos? ¿Por qué?

Posiblemente el individuo escogerá el código que le resulto más fácil de leer a nivel de las expresiones lógicas a pesar de ser equivalentes, el hecho está en que la pregunta pide escoger uno de los dos.

Para la pregunta de reflexión se espera que escojan el código D pues es más “fácil” de leer.

iii. ¿Encuentras alguna relación entre los códigos A y B que analizaste anteriormente con los códigos C y D?

Se espera que encuentren una relación entre las expresiones lógicas que tuvieron los códigos A y B con las expresiones que encontraron en los códigos C y D, además de que estas expresiones le son más familiares pues anteriormente estuvieron trabajando con ellas en la sección uno de las actividades.

Sección 3.

i. Completa las siguientes tablas.

Se espera que los individuos puedan completar la tabla como se muestra en la figura 5.13 siguiente.

A	B	A and B	not (A and B)	not A and not B
Verdad	Verdad	Verdad	Falso	Falso
Verdad	Falso	Falso	Verdad	Falso
Falso	Verdad	Falso	Verdad	Falso
Falso	Falso	Falso	Verdad	Verdad

A	B	A or B	not A or not B	not (A or B)
Verdad	Verdad	Verdad	Falso	Falso
Verdad	Falso	Verdad	Verdad	Falso
Falso	Verdad	Verdad	Verdad	Falso
Falso	Falso	Falso	Verdad	Verdad

Fig. 4.13

ii. ¿Qué relación pudiste encontrar entre las últimas dos columnas de la tabla A y la tabla B, encuentras alguna similitud?

Se espera que el individuo pueda relacionar que ambas tablas tienen similitud y si las tablas de verdad coinciden entonces argumente que son equivalentes en cuanto a la relación a cada columna.

iii. Relaciona la salida esperada de cada código con su respectiva tabla de verdad y anótalos en la siguiente tabla.

Se espera que los individuos puedan concretar la siguiente salida como se muestra en la tabla (Figura 5.14), valiéndose de la experiencia que produjeron los reactivos anteriores; esto es, el uso de los operadores lógicos: conjunción and (&&) , disyunción or (||) y la negación not (!).

TABLA	CODIGO
uno	tres
dos	cuatro
tres	uno
cuatro	dos

Fig. 4.14

4.1.3 Actividad 2

Sección 1.

i. Completa la tabla desarrollando para cada caso las leyes de De Morgan respectivamente.

Se espera que los individuos logren contestar tal y como esta expresado en la siguiente tabla (Figura 5.15).

A	B	$not(A\ and\ B)$	$not(A\ or\ B)$
$x < a$	$x >= b$	$x >= a\ or\ x < b$	$x >= a\ and\ x < b$
$x > b$	$x == b$	$x <= b\ or\ x\ != b$	$x <= b\ and\ x\ != b$
“Juan es cocinero”	“pedro es policía”	“juan no es cocinero” or “pedro no es policía”	“juan no es cocinero” and “pedro no es policía”
$\neg(R)$	$\neg(S)$	$R\ or\ S$	$R\ and\ S$
(R)	$\neg\neg(S)$	$notR\ or\ notS$	$notR\ and\ notS$
$a\ != b$	$a\ != c$	$a == b\ \ or\ a\ == c$	$a == b\ \ and\ a\ == c$

Fig. 4.15

Sección 2.

Se solicita analizar el siguiente código:

```
//CODIGO E
#include <stdio.h>
#include <iostream>
using namespace std;
int main()
{
      bool A[] = {true, false};
      bool B[] = {true, false};
      bool C[4] = {};
      bool D[4] = {};
      bool iguales = true;
      int contador=0;

          for(int x=0;x<2;x++){
              for(int y=0;y<2;y++){
                C[contador] = not(A[x] or B[y]);
                D[contador] = not(A[x]) and not(B[y]);
                cout<<C[contador]<<"|"<<D[contador]<<endl;
                   contador = contador + 1;
               }
          }
          for(int x=0;x<contador;x++){
                if(C[x] != D[x])
                      iguales = false;
                      break;
          }
          if(iguales)
                      cout << "se cumple la ley de Morgan" <<endl;
```

```
29.          else
30.                      cout << "No se cumple la ley de Morgan" <<endl;
31.      }
```

¿Cuál será la salida del código E?

Se espera que argumenten, “se cumple la una ley de Morgan”, mediante una inspección de código, puedan determinar que se está empleando la ley negación de la disyunción $\boldsymbol{not(a\ or\ b), not\ a\ and\ not\ b}$.

2. ¿Está presente alguna ley de De Morgan en el código E? señala qué líneas son las que evalúan esta ley.

Se espera que los individuos argumenten que se está empleando la ley $not(a\ or\ b), not\ a\ and\ not\ b$, (la segunda ley) y que las líneas que hacen que se ejecute dicha la ley son 16,17 y 23.

¿Alguna de las leyes no está presente? ¿Cuál?

Se espera que puedan dar la otra ley de De Morgan la cual es la negación de la conjunción $not(a\ and\ b) = not\ a\ or\ not\ b$

ii. Si respondiste que falta alguna ley en el programa anterior, ¿qué líneas cambiarias en el código para que se pueda ejecutar la otra ley de De Morgan?

Se espera que señalen que las líneas a cambiar son las 16 y 17 (figura 5.16). Así la nueva versión de su código quedaría de la siguiente manera.

```
1. for(int x=0;x<2;x++){
2.         for(int y=0;y<2;y++){
3.           C[contador] = not(A[x] and B[y]);
4.           D[contador] = not(A[x]) or not(B[y]);
5.              contador = contador + 1;
6.          }
7.     }
```

Fig. 4.16

iii. Completa la expresión de la línea 15 del código F para que el código pueda ejecutarse y se cumpla la expresión.

Se espera que en la línea 16 (Figura 5.17) puedan completar la siguiente expresión, la cual corresponde a la equivalencia de la ley que se muestra.

```
16.    D[contador] = R[x] or H[y];
```

Fig. 4.17

Sección 3.

i. Aplica las leyes de De Morgan a los siguientes ejercicios.

Se les proporciona una expresión y los individuos tienen que expresar su forma equivalente desarrollado mediante la implementación de las leyes de De Morgan.

1. $not(q_1 \; and \; q_2) \equiv (not \; q_1 \; or \; not \; q_2)$
2. $not(p_1 \; and \; p_2 \; and \; p_3) \equiv (not(p_1) \; or \; not \; (p_2) \; or \; not(p_3))$
3. $not(not \; (r_1) \; and \; not(r_2) \; and \; not(r_3)) \equiv r_1 \; or \; r_2 \; or \; r_3$
4. $not(s_1 \; or \; s_2 \; or \; s_3 \; or \; s_4) \equiv not(s_1) \; and \; not \; (s_2) \; and \; not(s_3) \; and \; not(s_4)$
5. $not(not(t_1) \; or \; not(t_2) \; or \; not(t_3) \; or \; not(t_4)) \equiv t_1 \; and \; t_2 \; and \; t_3 \; and \; t_4$

ii. Según tus respuestas de la parte anterior:

1. ¿Observas algún patrón en los ejercicios anteriores?

Una respuesta o argumento esperado seria que resaltaran que para cualquier número de proposiciones siempre se cumplirían las leyes de De Morgan.

2. ¿Es cierta la igualdad $\boldsymbol{not(p_1 \; and \; (p_2 \; or \; p_3))} \equiv \boldsymbol{not((p_1 \; and \; p_2) \; or \; p_3)}$?

En este reactivo intervienen ambas leyes, por lo que se espera que el individuo haya adquirido cierta madurez que fue adquiriendo en los reactivos anteriores, aquí la manera que tienen para argumentar su resultado es por medio de las herramientas ya sea por medio de tablas o que desarrollen la expresión para demostrar su invalidez.

iii.

1. Propón expresiones para la negación de la disyunción y la conjunción para el caso de $\boldsymbol{n}$ premisas de acuerdo a las leyes de De Morgan.

El resultado esperado es que proporcionen una respuesta similar o igual a la siguiente.

$$not(A_1 and \; A_2 \; and \; A_3 and \; \ldots \; and A_n) = (not(A_1) \; or \; not(A_2) \; or \; not(A_3) \; or \; \ldots \; or \; not(A_n))$$

$$not(A_1 or A_2 \; or \; A_3 or \; \ldots \; or A_n) = (not(A_1) \; and \; not(A_2) \; and \; not(A_3) \; and \; \ldots \; and \; not(A_n))$$

2. Explica verbalmente qué significan esas expresiones.

Se pretende que los individuos identifiquen el desarrollo de las leyes, al emplear el operador sobre la disyunción y la conjunción, a lo que podrían argumentar que: “la negación de un conjunto de premisas unidas por el operador lógico and es equivalente a la negación de cada premisa unidos ahora por el operador lógico or, y lo mismo para el otro caso.

4.2 Análisis a priori bajo al marco APOE

Con motivos de que las actividades se realizaron con el propósito de favorecer algunas de las construcciones mentales que marca el marco APOE, en esta sección daremos una explicación sobre que concepción se pretendió en cada actividad y sus respectivas secciones.

4.2.1 Actividad 0

Como hemos mencionado con anterioridad, la actividad cero fue realizado con actividades de diagnóstico y exploración que nos brindara información si es que los individuos cuentan con las construcciones previas, sin embargo, forman parte de las actividades por tanto lo analizaremos de igual manera desde el punto de vista del marco teórico.

Sección 1.

Lo que se pretende con esta actividad es trabajar totalmente a nivel Acción, se pretende que los individuos se familiaricen con las desigualdades sobre la recta; por lo tanto, los individuos manipulan las herramientas. Estas actividades se caracterizan por fomentar la repetición, con lo cual el propósito de este primer ejercicio es llevarlo a un nivel próximo a la interiorización.

Sección 2.

Una de las herramientas que van a ocupar en el resto de la actividad es la familiaridad con los conceptos de lógica; su notación y simbología, es por ello que aparecen en esta primera parte. El trabajo que se realiza es a nivel de concepción Acción, retomando lo que ya se trabajó e incorporando los operadores lógicos, fomentando preguntas que requieren razonamiento.

Sección 3.

Aquí tal vez el cambio pareciera un poco más drástico y fuera de una secuencia, pero no es así. Los ejercicios propuestos a nivel de código requieren algo más que Acción, es por ello que se requiere que los estudiantes ya cuenteen con la concepción de Proceso sobre el lenguaje de programación. Debido a que se estuvo trabajando a nivel Acción en los ejercicios anteriores se espera que al menos adquirieran por medio de la interiorización el Proceso a nivel desigualdad y operador lógico, y se espera que puedan coordinarlo con el Proceso de lenguaje de programación, y de esta manera puedan resolver el ejercicio.

Sección 4.

En esta última parte de la actividad cero, se trabaja a un nivel de Proceso. Aquí se tiene que efectuar las operaciones lógicas de la negación y de igual manera la habilidad de relacionar e imaginar el valor resultante sin tener la recta como referencia. Esto es posible lograrlo a partir de la interiorización y el trabajo efectuado de los ejercicios anteriores.

4.2.2 Actividad 1

Sección 1.

A lo largo de esta actividad se llevará a cabo una serie de ejercicios, los cuales están orientados a favorecer la construcción mental Acción que es la que marca el marco teórico APOE. Retomando que una acción consiste en una transformación de un objeto que es percibida por el individuo como externa y se realiza como una reacción a sugerencias que proporcionan detalles de los pasos a seguir (Arnon et al., 2014).

De tal manera, se pretende observar el uso de concepciones Acción al solicitar que los estudiantes realicen manipulaciones específicas a objetos previos, como: El manejo de las tablas de verdad, la expresión de desigualdades sobre la recta, representación de algoritmos de decisión con expresiones lógicas y por último la lógica de predicados.

Caso 1: $\boldsymbol{not(P \wedge Q) = \; not(P) \vee not(Q)}$

En esta primera sección de la actividad se pretende favorece la concepción Acción, pues el individuo cuenta con entradas fijas. Estas corresponden a las desigualdades y operadores lógicos que debe manipular empleando las hojas de trabajo asignadas para obtener su representación en la recta, la cual corresponde a los pasos que tiene que realizar y por último que logren una asimilación por medio de una comparativa lo que llamamos una salida esperada.

Este ejercicio tiene como objetivo la manipulación sobre las gráficas, además de distinguir los valores numéricos para poder representarlos sobre la recta. Se empleará además la debida distinción entre las operaciones sobre las desigualdades y los operadores lógicos.

i y ii. Representa las siguientes expresiones y Compara ambas gráficas y escribe qué relación encontraste en ellas.

La concepción de acción se logra mediante varias etapas o pasos los cuales se pueden reflejar en una serie de ejercicios que tienen un seguimiento, para el caso de los incisos i y ii, se está trabajando los primeros pasos de acción.

Aquí el estudiante debe de relacionar las desigualdades que estuvo trabajando a través de una comparación. Las entradas de la acción en este caso son las desigualdades y los operadores lógicos, la salida esperada es generar una conclusión o una reflexión sobre los datos y las operaciones que realiza. En este punto todavía los estudiantes no se espera que puedan adquirir la concepción acción aun, pretendemos que los problemas siguientes les ayuden a terminar de construir la acción y que ellos decidan aplicar los pasos sin la necesidad de darles explícitamente las instrucciones dictadas.

iii. llenado de la tabla para $(\boldsymbol{P \wedge Q})$.

Se pretende que al usar el ambiente dinámico los estudiantes puedan comparar sus procedimientos y resultados del ejercicio anterior.

Continuando con las etapas de la acción, en este caso particular, y con ayuda del ambiente dinámico se busca que los individuos empiecen a generar resultados mediante la observación que obtienen, el cual pueden lograr mediante la mecanización el cual comienzan sin la necesidad de recordarles continuamente el ingreso de los valores P y Q.

De esta manera se logran concebir todos los pasos de acción, los cuales son trabajar con las entradas, operar sobre ellas y obtener una salida.

iv. ¿Qué observaste en las gráficas que obtuviste al ingresar los valores?, ¿tuviste alguna complicación?

Con la manipulación de los valores se pretende que los individuos puedan estar mecanizar simultáneamente la aplicación de los operadores lógicos conjunción y negación los cuales forman parte de los objetos de entrada, generando así una acción con ellos.

Es posible que el resultado de este tipo de preguntas arroje resultados y afirmaciones universales mismas que podrían relacionarse con el inicio de la concepción Proceso.

v. ¿El applet calculó bien los valores de R?, ¿Qué usaste para decidir tu respuesta?

Este ejercicio funciona para que los estudiantes practiquen los pasos correspondientes a la concepción Acción. De lo que disponen son las herramientas con las que estuvieron trabajando previamente o bien las operaciones que realizan de manera mental.

vi. ¿Qué observaste en las gráficas que obtuviste al ingresar los valores?, ¿tuviste alguna complicación? vii. ¿Hay algo que siempre se cumpla al ingresar los valores en la tabla?

Se pretende que, al hacer una revisión sobre ambas salidas, **R** y **not(R)**, se mecanice la ejecución de los pasos. Esto es importante pues se trabaja siempre la primera igualdad. Se favorece la reflexión sobre la propiedad invariante ya que esto favorece la reflexión sobre el objeto de la primera ley.

Se pretende que puedan reflexionar sobre que "la propiedad negación del and es el negado del or ", y de esta manera se eviten las complicaciones comunes.

viii. ¿La expresión $\boldsymbol{not(P \wedge Q) = notP \wedge notQ}$ es posible emplearla?, pruébalo en el applet y después escribe tus observaciones

Aquí ya pueden tener más herramientas para poder llegar a una conclusión partiendo de las concepciones ganadas anteriormente. Aunque aquí se pretende reflexión sobre todo el trabajo que realizaron anteriormente, se espera además que puedan distinguir que la regla asociativa de la negación en el caso de los operadores lógicos se emplea distinto a la regla distributiva en algebra ejemplo: $a(b+c) = ab+ac$.

Para este reactivo en particular se logra visualizar la necesidad de pensar a un nivel superior de la concepción Acción, es por ello que simplemente una respuesta negativa es suficiente en este caso.

Caso 2: $\boldsymbol{not(P \vee Q) = not(P) \wedge not(Q)}$

Aquí se pretende favorece la concepción Acción de igual manera como se hizo en el caso a), pues el individuo cuenta con entradas las cuales son las desigualdades y los operadores lógicos, sobre los cuales este debe manipular sobre la gráfica para obtener una salida esperada, lo cual tiene como objetivo favorecer la generalización de las proposiciones.

i,ii. Representa las siguientes expresiones en su grafica correspondiente y compáralas.

Aquí la tarea al igual que en el caso a) es que el estudiante debe relacionar las distintas desigualdades a través de una comparación. Las entradas de la acción en este caso son las desigualdades y la salida esperada es generar una conclusión o una reflexión sobre los datos y las operaciones que realiza. En este punto todavía no se espera que los estudiantes puedan adquirir la concepción acción: sin embargo, sí pretendemos que estos ejercicios les puedan servir para los problemas siguientes.

La principal diferencia es el cambio del operador, pero sigue contando con las mismas características que corresponden a la concepción de Acción.

iii. llenado de la tabla para $(\boldsymbol{P} \vee \boldsymbol{Q})$.

Se pretende que al usar el ambiente dinámico puedan comparar los resultados del ejercicio anterior y también puedan adquirir la repetición, pero sin descuidar la comparación, que es lo que los llevará a adquirir los primeros pasos de la acción.

iv,v. ¿El applet calculó bien los valores de R?, ¿qué usaste para decidir tu respuesta?

Se pretende que al estar haciendo una revisión sobre ambas salidas la cual lleva la ejecución de los pasos, las cuales se podrán tomar como entradas las cuales deben de generar un patrón constante y posteriormente operar sobre ellas haciendo la labor de compararlas, pueda manejar la concepción de desigualdad y de igual manera sobre los operadores lógicos $and, or\ y\ not$, se favorece la reflexión sobre la propiedad invariante.

vi. ¿Hay algo que siempre se cumpla al ingresar los valores en la tabla?

Se pretende que puedan reflexionar sobre que "la propiedad negación del and es el negado del or ", de esta manera se evitan las complicaciones comunes.

vii. ¿La expresión $\boldsymbol{not}(\boldsymbol{P} \vee \boldsymbol{Q}) = \boldsymbol{notP} \vee \boldsymbol{notQ}$ es posible el poder emplearla?, pruébalo en el applet y después escribe tus observaciones.

Aquí ya pueden tener más herramientas para poder llegar a una conclusión partiendo de las concepciones ganadas anteriormente, aunque aquí de igual manera se pretende reflexión sobre todo en el trabajo que realizaron anteriormente; se espera además que puedan distinguir que la regla distributiva de la negación en el caso de los operadores lógicos, se emplea distinto a la regla distributiva en algebra; también se puede comparar con el problema del binomio al cuadrado, en donde se suele cometer el error de desarrollar la expresión como $(a+b)^2 = a^2 + b^2$ cuando esto es incorrecto.

viii. ¿Qué relación encuentras entre los resultados del caso a) y los resultados que obtuviste en el caso b)?

Se espera que se empiecen a relacionar las leyes de De Morgan, para después coordinarlas como procesos.

Sección 2.

Para esta sección mucho del contenido es expuesto de manera similar al anterior, pues es evidente que igualmente lo que se requiere es trabajar la concepción Acción; la diferencia es que en estos ejercicios se puntualiza el cambio del contexto. El propósito es que los estudiantes puedan construir y trabajar el concepto así como la concepción en cualquier ámbito.

En las hojas de trabajo se sugiere que los estudiantes trabajen la resolución de los códigos en papel o de manera mental; parte de ello nos ayudara a saber si es que los individuos manejan la concepción de Proceso y Acción de los ciclos simples y los ciclos anidados, Cetin (2015).

a) Primer caso: $\boldsymbol{not}(\boldsymbol{P} \wedge \boldsymbol{Q}) = \boldsymbol{not}(\boldsymbol{P}) \vee \boldsymbol{not}(\boldsymbol{Q})$

i. Ejecuta los códigos A y B, usa el diagrama para ayudarte y ve marcando sobre él, las iteraciones que creas conveniente.

ii. ¿Qué es lo que están haciendo los códigos A y B? explica su salida.

Se cuenta con las entradas las cuales son los algoritmos de los códigos A y B y mediante el estar operando sobre ellos y sus correspondientes gráficas, se fomenta una acción sobre ellos, pues a través de una secuencia de pasos se pretende lograr la salida, la cual es el diagrama de la partida.

Mediante la comparación de los resultados se fomenta la reflexión de lo que estuvieron operando en el ejercicio anterior y que además puedan distinguir la equivalencia de la condición lógica que se encuentra dentro del ciclo for de los códigos A y B.

Una de las características a resaltar para este tipo de ejercicios es que no se orienta al individuo a que genere ciertos pasos, pues se espera que ellos mismos los puedan generar; el motivo de esto es que el estudiante asimila este ejercicio como los anteriores en los que ya estuvo trabajando, pues si bien se está manejando otro contexto, el concepto y las herramientas son las mismas que ha trabajado desde el comienzo.

iii. ¿Crees que es conveniente para la maquina el haber preferido utilizar alguno de los dos códigos? ¿Por qué?

Se pretende reflexión sobre las expresiones pues ambas lo condujeron hacia un mismo resultado, se espera que logren comparar una vez más los resultados obtenidos anteriormente para que de esta manera puedan reforzar la relación de las expresiones.

iv. Para el código B, ¿si quitamos la condición $!(\boldsymbol{Peon}[\boldsymbol{i-1}][\boldsymbol{0}] != \boldsymbol{NULL}$ dentro del ciclo $\boldsymbol{while}$, consideras que se pueda seguir ejecutando el juego para ese caso particular? ¿Por qué?

Se espera que mediante la ejecución nuevamente del código, logren esta vez la notable diferencia entre el código A y el código B, a su vez que distingan entre la diferencia de las expresiones a pesar de sus equivalencias.

b) Segundo caso: $\boldsymbol{not}(\boldsymbol{P} \vee \boldsymbol{Q}) = \boldsymbol{not}(\boldsymbol{P}) \wedge \boldsymbol{not}(\boldsymbol{Q})$

i. ¿Cuál es el resultado que obtuviste con los códigos C y D?

La acción se trabaja a través de la operación de los códigos y la salida la pueden estar realizando en el diagrama y una vez terminado pueden hacer la comparación del diagrama resultante con el diagrama del ajedrez que se planteó al inicio.

ii. ¿Crees que es conveniente para la maquina el haber preferido utilizar alguno de los dos códigos? ¿Por qué?

Se pretende reflexión sobre las expresiones pues ambas lo condujeron hacia un mismo resultado, se espera que logren comparar una vez más los resultados obtenidos anteriormente para que de esta manera puedan reforzar la relación de las expresiones.

iii. ¿Encuentras alguna relación entre los códigos A y B que analizaste anteriormente con los códigos C y D?

A partir del análisis sobre el trabajo que estuvieron realizando anteriormente y este nuevo, se pretende una reflexión sobre la notación y el manejo de los operadores lógicos, debido a que a pesar de la pequeña variación en la sección de la condición, la aplicación de la regla sigue siendo equivalente.

Sección 3.

i. Completa las siguientes tablas.

Se fomenta acción mediante la observación sobre las columnas de las tablas y las columnas que se piden que deban llenar.

ii. ¿Qué relación pudiste encontrar entre las últimas dos columnas de la tabla A y la tabla B, encuentras alguna similitud?

Las salidas del ejercicio anterior servirán para las entradas en este caso, y a través de la reflexión lograr la relación de las expresiones o lo que son el comienzo de las leyes de De Morgan pues es la primera vez que se muestran en notación de lógica.

iii. Relaciona la salida esperada de cada código con su respectiva tabla de verdad y anótalos en la siguiente tabla.

Cada entrada es correspondiente a cada código y cada salida es correspondiente a cada tabla, y mediante la operación de cada uno de ellos se logrará obtener cada salida, los códigos no ayudan de momento a las leyes de Morgan, sino más bien están pensados para poder encaminarlos hacia las siguientes actividades en las cuales estarán manejando unos problemas similares, solo que en este caso ya se pretende que tengan proceso.

4.2.3 Actividad 2.

En la actividad uno se estuvo trabajando con la concepción Acción, y se presentaron algunos ejercicios que favorecían la reflexión sobre la repetición y la manipulación sobre elementos que proporcionaban salidas, aquí en esta nueva actividad estaremos favoreciendo la concepción proceso, que es de acuerdo a como lo señala el marco teórico APOE.

Se habla de un Proceso cuando las acciones se repiten y esto lleva a que el individuo pase de confiar en señales externas a tener control interno sobre ellas, lo cual se caracteriza por adquirir la capacidad de imaginar y llevar a cabo los pasos sin tener que realizar cada operación explícitamente y poder omitir los pasos. Esta es la manera en que se establece el paso de acción a proceso y se logra mediante el mecanismo mental de interiorización.

Un Proceso es cuando un individuo ha interiorizado la concepción de Acción. En la concepción Acción el individuo genera una serie de pasos que requiere para poder llegar a una solución, en un Proceso el individuo es capaz de generalizar los pasos; en otras palabras, el individuo puede realizar estos pasos mentalmente es decir se puede imaginar la salida al aplicar una operación específica sobre un objeto matemático.

Como ya se mencionó esta actividad corresponde a favorecer la concepción de Proceso, por lo tanto proporcionaremos una descripción general de lo que esto implica en el sentido de las actividades. La cualidad y forma de pensamiento para los que inician esta actividad dos, es que los individuos ya cuentan con herramientas o construcciones que fueron elaborando durante las secciones de la actividad uno, es decir están en la etapa en la que la concepción Proceso se desarrolla. También se ha mencionado que adquirir un Proceso no es una tarea fácil, por lo que en las primeras secciones de la actividad dos se verá la necesidad de resaltar si es que tales concepciones se han adquirido. Una de las diferencias notables que tienen estas actividades con respecto a las anteriores es que en este apartado las preguntas además de corresponder a un enfoque dirigido al concepto matemático y a sus aplicaciones poseen la cualidad de que la respuesta no se genera de forma independiente, es decir, si bien estas generan la propiedad de la reflexión de igual manera, son los pasos para llegar a la respuesta los que no son necesarios en este punto. Por ejemplo, para el caso de la concepción Acción se requirió que las actividades contaran con los tres elementos que la caracterizan los cuales son; elementos de entrada, las operaciones sobre estos elementos y los resultados obtenidos o elementos de salida, así bien la concepción de Proceso para esta sección plantea que los individuos puedan avanzar sin la necesidad de esos pasos.

Sección 1.

i. Completa la tabla desarrollando para cada caso las leyes de De Morgan respectivamente.

De acuerdo con la concepción proceso, los individuos ya no tendrían que hacer todas las operaciones, que es lo que realizaban en las actividades uno cuando estaba trabajando con proceso, aquí se espera que puedan dar directamente la respuesta, pues esa sería una evidencia de proceso, el individuo debe de ser capaz de realizar las operaciones en su mente imaginando la salida esperada sin hacer todos los pasos, lo que genera una reflexión.

Sección 2.

A continuación, Analiza el siguiente código.

i. ¿Cuál será la salida del código E?

Aquí los individuos tienen que correr el código en su mente para obtener la salida, la cual es una característica de proceso, además de eso tienen que trabajar la expresión de la ley de Morgan el cual los llevaría a otro proceso pues el individuo esta imaginando la salida.

¿Está presente alguna ley de De Morgan en el código E? señala qué líneas son las que evalúan esta ley.

El individuo manipula el código, y lo divide en secciones y da prioridad a las líneas de código que son más relevantes, el cual es un indicativo que está tomando control sobre las acciones, el cual es un indicativo de que están separando acción de proceso.

¿Alguna de las leyes no está presente? ¿Cuál?

Debido a que en este punto los individuos tienen un acercamiento a la interiorización de acción de las leyes de Morgan, pueden darse cuenta de que las leyes de Morgan vienen siempre en parejas, por lo que puedan escribir la otra expresión sería un indicio de que los individuos están manejando un Proceso.

ii. Si respondiste que falta alguna ley en el programa anterior, ¿qué líneas cambiarias en el código para que se pueda ejecutar la otra ley de De Morgan?

La manipulación de código es evidencia de que la etapa de acción se ha completado y ahora pueden proponer nuevo código y además emplear directamente la ley de De Morgan a la expresión lo cual implica que se está efectuando un Proceso sobre la expresión.

Escribe tu versión de esas líneas, (puedes escribir solamente las líneas que consideres):

Si el individuo es capaz de generar código y además de expresar el resultado esperado de las leyes, quiere decir que el individuo está generando la concepción de Proceso, pues el individuo puede ver las leyes como formulas; además de que el individuo no tendrá que correr el código pues se el individuo se imagina la salida que obtendrá sin tener que correr el código.

iii. Completa la expresión de la línea 15 del código F para que el código pueda ejecutarse y se cumpla la expresión.

Este reactivo expresa la concepción Proceso porque está proponiendo la expresión equivalente empleando directamente las leyes de De Morgan como fórmulas, sin tener que realizar ningún paso para llegar a ella. En otras palabras, ya tiene interiorizadas las leyes.

Sección 3.

Para esta sección, si bien el objetivo principal fue el desarrollar actividades en donde el nivel de complejidad en los reactivos fuera incrementando, también lo fue planeado para el nivel de concepción mental, inicialmente se pensaron las actividades para que fueran parte de la concepción Proceso; sin embargo mediante la revisión de los reactivos vimos que estos requerían algo más que los Procesos simples. Esto se puede llevar a cabo ya sea por medio de implementar un cambio de variable o la misma coordinación de Procesos simples, lo que nos genera como resultado los Procesos avanzados. Ejemplo de ello es utilizar la ley de la doble negación para un término o bien aplicar una ley a una expresión que contenga más de dos proposiciones y que a su vez este contenga expresiones negadas en contextos ya sea en formato lógico simbólico o en enunciado del lenguaje natural.

i. Aplica las leyes de De Morgan a los siguientes ejercicios.

En este reactivo se evidencia la concepción de Proceso debido a que el individuo aplica ambas leyes como formulas sin tener que hacer ningún paso para corroborar que su resultado es correcto, el individuo supone que llega al resultado correcto debido a la experiencia que tuvo anteriormente en la concepción Acción.

ii. Según tus respuestas de la parte anterior:

1. ¿Observas algún patrón en los ejercicios anteriores?

El reactivo genera una reflexión por parte del individuo, esto con propósito de que pueda visualizar la generalización de las leyes de De Morgan y que sin importar el número de proposiciones estas siempre se cumplirán, por tanto el individuo tiene que imaginar que la propiedad siempre se cumplirá, por tanto es evidencia de Proceso.

2. ¿Es cierta la igualdad $\boldsymbol{not}(\boldsymbol{p_1}\ \boldsymbol{and}\ (\boldsymbol{p_2}\ \boldsymbol{or}\ \boldsymbol{p_3})) \equiv \boldsymbol{not}((\boldsymbol{p_1}\ \boldsymbol{and}\ \boldsymbol{p_2})\ \boldsymbol{or}\ \boldsymbol{p_3}))$?

Este reactivo genera Proceso porque se espera que los individuos imaginen maneras de llegar a que la solución es incorrecta. Esto lo pueden lograr de diferentes maneras, una de ellas es aplicar las leyes directamente como formulas sobre la expresión y encontrar que no son equivalentes, otra manera es que desarrollen la tabla de verdad; sin embargo esta última es posible que no ayude a construir la concepción de Proceso, pues la generación de una tabla de verdad supone una serie de pasos lo cual corresponde claramente a una concepción de Acción.

iii.

En este caso estas dos preguntas vienen de manera conjunta, pues en la primera generan la expresión y en la segunda se pide una explicación de la explicación que generaron

1. Propón expresiones para la negación de la disyunción y la conjunción para el caso de $\boldsymbol{n}$ premisas de acuerdo a las leyes de De Morgan.

2. Explica verbalmente qué significan esas expresiones.

Estos reactivos proponen Proceso, pues el mismo reactivo sugiere que se piense para el caso generalizado de las leyes de De Morgan. No es necesario ofrecer una demostración rudimentaria, sin embargo, sí se espera que el individuo pueda darse cuenta en la generalización de que las leyes se cumplen para cualquier número de proposiciones.

5 Análisis de datos y presentación de resultados

5.1 Implementación de las actividades

Para la puesta en escena se escogieron estudiantes de la licenciatura en Ciencias de la Computación del departamento de matemáticas de la universidad de sonora dado que cumplen el perfil definido en nuestra propuesta.

La implementación de las actividades se llevó en conjunto en dos grupos de estudiantes los cuales algunos estaban cursando simultáneamente el curso de *Programación de Computadoras* y el de *Matemáticas Discretas*, estos se ofrecen el segundo semestre de la carrera de Ciencias de la Computación en el segundo semestre.

El motivo de elegir estos estudiantes en ese semestre en particular es que es en esos cursos donde aparece por primera vez los temas que corresponden a nuestro estudio.

La intervención didáctica se llevó a cabo en cuatro sesiones de una hora. A los participantes se les facilitaron unas hojas de trabajo. En un laboratorio de cómputo, ellos podían seguir un enlace que los dirigía a una página web (www.mat.uson.mx/~teoriadeconjuntos/) que es complementaria a las hojas de trabajo mencionadas.

Durante las sesiones a los participantes se les proporcionaron instrucciones de cómo debían seguir las actividades y se atendieron dudas particulares como el manejo de la página web y los applets. En todas las sesiones, los estudiantes estuvieron trabajando de manera individual.

Durante la revisión de las hojas de trabajo de varios estudiantes se pudieron notar algunas dificultades persistentes; es decir, presentaron dificultad en algún ejercicio específico y el error lo repitieron durante el resto de la actividad. En varios casos fue notable que las propiedades de las leyes de De Morgan pudieron aplicarse a contextos dentro de la lógica a nivel abstracto, pero al cambiar del contexto –de proposiciones lógicas a un contexto de aplicación en un lenguaje de programación–, se percibieron dificultades que no habían presentado antes, lo cual indica que no pudieron aplicar las leyes de De Morgan a estos otros contextos.

5.2 Análisis de las concepciones de las leyes de De Morgan

En esta sección mostraremos algunos de los resultados obtenidos durante la puesta en práctica; particularmente se retomarán los casos de éxito. Estos son aquellos donde el individuo no mostró rasgos de complicación durante la realización de las actividades y en cambio hubo indicativos de que se logró un aprendizaje.

5.2.1 *Acción Primera Ley*

Definición Acción Primera Ley $\neg(\boldsymbol{p} \wedge \boldsymbol{q})$

La concepción de Acción para las leyes de De Morgan consiste de tres elementos los cuales son: objetos de entrada, transformación de las entradas por medio de operaciones, y el resultado de una comparación como objeto de salida; estos elementos son independientemente de los contextos que se tomen:

1. Objetos de entrada:
   1.1. Un par de proposiciones $(\boldsymbol{p}, \boldsymbol{q})$ y sus negaciones
2. Pasos/operaciones sobre los objetos:
   2.1. Calcular la conjunción $\boldsymbol{p} \wedge \boldsymbol{q}$
   2.2. Calcular la negación de la conjunción $\neg(\boldsymbol{p} \wedge \boldsymbol{q})$
   2.3. Calcular la disyunción de las negaciones de los elementos $\neg\boldsymbol{p} \vee \neg\boldsymbol{q}$, y si es necesario previamente calcular las negaciones $\neg p$, $\neg q$.
   2.4. Comparar $\neg(\boldsymbol{p} \wedge \boldsymbol{q}), \neg\boldsymbol{p} \vee \neg\boldsymbol{q}$
3. Objetos de salida: El elemento final de la concepción Acción es la relación entre los elementos que obtuvo en 2.2 y 2.3, lo que propicia que se logre una reflexión sobre la equivalencia $\neg(p \wedge q) = \neg p \vee \neg q$, es decir que *la negación de una conjunción es equivalente a la disyunción de las negaciones*.

Para esta parte se presentan diferentes contextos, que se fueron presentando durante las actividades, se desea mostrar que los estudiantes sí utilizan la concepción descrita arriba. Llamaremos casos de éxito a las respuestas de los estudiantes que cumplen los pasos de la concepción Acción.

Casos de éxito

Esta concepción se encuentra en los problemas de la Actividad 1, se mostrarán respuestas de distintos estudiantes para cada contexto.

Contexto 1: Desigualdades

Este contexto se aborda en la sección 1, caso a). Posterior a la implementación de las actividades se notó que las gráficas del inciso *i* están en orden opuesto a lo señalado arriba, se analizarán las implicaciones de esto posteriormente.

Se presenta el caso del estudiante **EC**, Él se toma como ejemplo porque hizo explícitas todas las partes de sus respuestas. Este estudiante solo alcanzo la resolución de este contexto, es decir, solo pudo responder esta sección de la primera actividad posiblemente por deficiencias en el área de programación. Por lo anterior, no tenemos una descripción completa de la concepción Acción para este estudiante, debido a que es necesario tener todos los contextos.

1. Objetos de entrada:

El estudiante **EC** a partir de interpretar las instrucciones y avanzar en las respuestas, identificó las proposiciones p y q y los operadores lógicos. Por consiguiente, para él las proposiciones son las siguientes desigualdades: $p: x \geq 10$, $q: x \leq 20$, $\neg p: x < 10$, $\neg q: x > 20$.

2. Pasos/operaciones sobre los objetos:
   2.1. El estudiante **EC** calcula primero $\boldsymbol{p} \wedge \boldsymbol{q}$, la expresión que está dentro del paréntesis. Este grafica en color negro (Figura 6.1), un segmento sobre la recta numérica, con marcas para los extremos, discutiremos después el uso de representaciones.

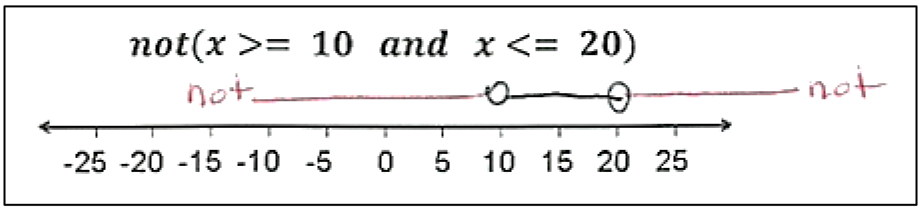


Fig. 5.1

   2.2. Para $\neg(\boldsymbol{p} \wedge \boldsymbol{q})$**,** el estudiante **EC** expresa sobre la misma gráfica (Figura n) en color rojo los elementos que no pertenecen al intervalo que obtuvo inicialmente, para hacer explícito que se trata de la negación, añade la etiqueta "not" a cada lado de la gráfica.
   2.3. El estudiante **EC** calcula la conjunción $\neg \boldsymbol{p} \vee \neg \boldsymbol{q}$ sobre la recta: graficó dos segmentos sobre la recta numérica, con marcas para indicar que no se incluyen los extremos (Figura 6.2). Aquí el estudiante **EC** pudo percatarse, mediante los resultados que obtuvo en la gráfica anterior, que $x < 10$ y $x > 20$ son las negaciones de $x \geq 10$ y $x \leq 20$.

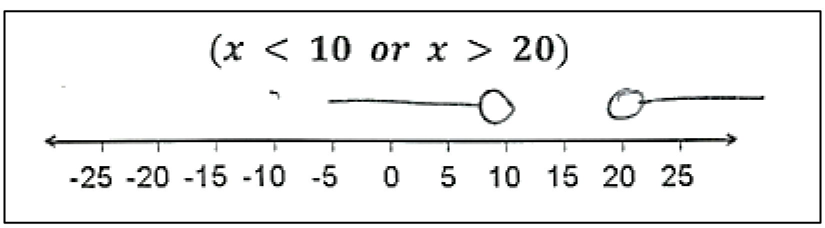


Fig. 5.2

   2.4. Comparar las proposiciones $\neg(\boldsymbol{p} \wedge \boldsymbol{q}) = \neg \boldsymbol{p} \vee \neg \boldsymbol{q}$ **,** esto se logra mediante la pregunta que se plantea en el reactivo ii (Figura 6.3). De esta manera el estudiante **EC** concluye que las expresiones $(x < 10 \; or \; x > 20)$ y $not(x >= 10 \; and \; x <= 20)$ son equivalentes, a través de la reflexión sobre sus respuestas del inciso *i.*

ii. Compara ambas gráficas y escribe que relación encontraste entre ellas.

(x < 10 or x > 20) = not (x >= 10 and x <= 20)

Fig. 5.3

3. Objetos de salida: Como era esperado el estudiante **EC** concluye la relación de equivalencia correcta y además la expresa en notación formal.

Contexto 2: Trabajo con el Applet.

Se pretende en esta etapa de la actividad, que al usar el ambiente dinámico los estudiantes puedan comparar sus procedimientos y resultados del ejercicio anterior.

Para este contexto, el estudiante **EA** finalizo todas las actividades, en el momento de la puesta en práctica de las actividades pudimos percatarnos que se trataba de un estudiante de quinto semestre y que estaba cursando por segunda ocasión los cursos de programación de computadoras y matemáticas discretas. Sin embargo, durante el trabajo realizado en la sesión el estudiante mostró algunas dudas y dificultades que estaban relacionadas con los códigos, específicamente en el contexto del ajedrez.

1. Objetos de entrada:

El estudiante **EA** identificó las proposiciones que estaban contenidas en las columnas P y Q de la tabla (véase Figura 6.4), mismos que debían de ser ingresados en el applet.

Caso	P	Q	R	$not(R)$
1	$x > 10$	$x \leq 20$	$(x > 10) \wedge (x \leq 20)$	$not(x > 10) \vee not(x \leq 20)$
2	$x > 5$	$x \leq 9$	$(x > 5) \wedge (x \leq 9)$	$not(x > 5) \vee not(x \leq 9)$
3	$x > -3$	$x \leq 12$	$(x > -3) \wedge (x \leq 12)$	$not(x -3) \vee not(x \leq 12)$
4	$x \geq 0$	$x \leq 9$	$(x \geq 0) \wedge (x \leq 9)$	$not(x \geq 0) \vee not(x \leq 9)$

Fig. 5.4

2. Pasos/operaciones sobre los objetos:
   2.1. las operaciones para calcular ($\boldsymbol{R}$) fueron procesadas por el ambiente dinámico, sin embargo, el individuo tuvo que trabajar en ir asignando el reactivo correspondiente sobre el applet que proporcionaba la expresión conveniente.

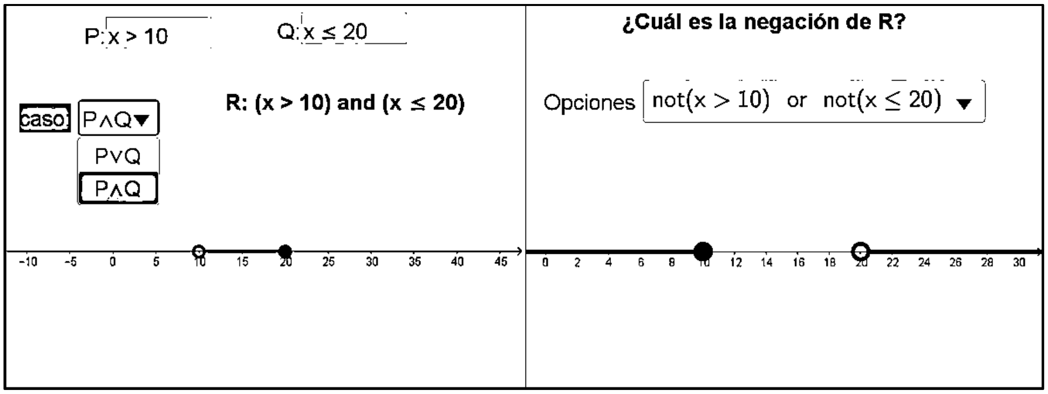
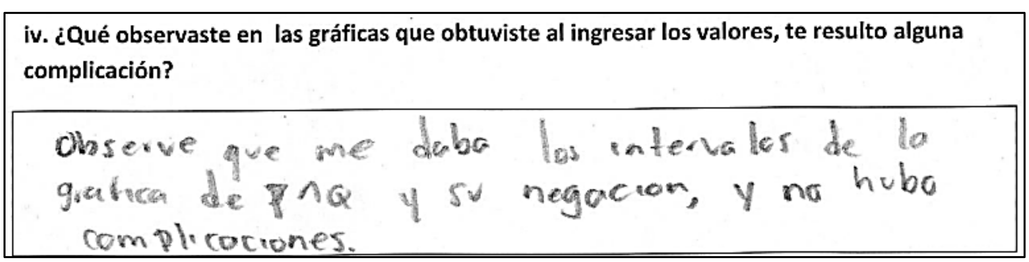


Fig. 5.5

2.2. En las operaciones para calcular $\neg(\boldsymbol{R})$ también fueron procesado internamente por el ambiente dinámico, y quedo a cargo del alumno comparar la expresión que se generó en $(\boldsymbol{R})$ con su correspondiente $\neg(\boldsymbol{R})$ de la lista desplegable de la parte derecha del applet, este contexto pretendía agilizar el contexto de desigualdad visto anteriormente. Por ejemplo, en el caso 1 de la tabla, el estudiante seleccionó correctamente $not(x > 10)\ or\ not(x \leq 20)$ (figura 6.5).

2.3. El estudiante **EA** determina que la opción correcta es la disyunción de las negaciones y lo expresa en la tabla de las hojas de trabajo basándose en las representaciones del ambiente dinámico.

2.4. El estudiante **EA** compara los elementos obtenidos del llenado de la tabla (Figura 6.4) y los resultados de la gráfica obtenidos por el applet. Se pretende que, al usar el ambiente dinámico, puedan adquirir la repetición, pero sin descuidar la comparación. Observamos que el individuo concluye que la negación de una conjunción coincide con la disyunción de las proposiciones negadas (figura 6.6).


iv. ¿Qué observaste en las gráficas que obtuviste al ingresar los valores, te resulto alguna complicación?

Observe que me daba los intervalos de la grafica de P∧Q y su negacion, y no hubo complicaciones.

Fig. 5.6

3. Objetos de salida:

En la respuesta mostrada arriba, el estudiante **EA** señala que, al aplicar la negación a un operador lógico, para el caso de una conjunción, se niegan las expresiones contenidas y el cambio del operador lógico and por una disyunción or.

Contexto 3: Ajedrez (código)

Este contexto es más complejo que los dos anteriores pues aquí se desea favorecer la concepción Acción dentro de ciclos de programación, por lo que es posible que las respuestas que se presentes tengan indicios de la concepción Proceso de ciclo como los describe Cetin(2015).

Presentamos las respuestas del estudiante **EF**, quién completó todas las actividades. Durante las sesiones de trabajo se le oriento particularmente en la interpretación de la mayoría de los códigos de programación, por lo que podemos concluir que dentro de las construcciones previas existen algunas deficiencias menores.

1. Objetos de entrada:

Para este contexto los elementos de entrada son proposiciones que se encuentran dentro de bucles de repetición de los códigos A (Figura 6.9) y B (Figura 6.10); donde el manejo de las desigualdades y los operadores lógicos siguen prevaleciendo. El estudiante **EF** identifica las proposiciones p y q en la condición del bucle de repetición (línea 9 Figuras 6.9,6.10) y en el índice i, de esta manera obtiene que: p: $Peon[i][0] == \; Peon == [i][0]$ y q: $Peon == [i - 1][0] \,! = \; NULL$.

2. Pasos/operaciones sobre los objetos:

   El estudiante **EF** calcula $(\boldsymbol{p} \wedge \boldsymbol{q})$

   2.1. y 2.2 En el código A el estudiante **EF** identifica $\boldsymbol{p} \wedge \boldsymbol{q}$ por lo que procede con las iteraciones del ciclo, las operaciones realizadas nos dan indicios que de alguna manera el estudiante aplico la negación a la expresión mientras ejecutaba el código, generando las salidas de cada iteración como se muestra en la Figura 6.7.

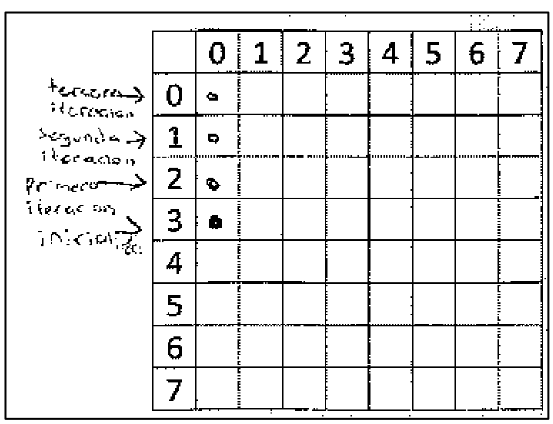


Fig. 5.7

2.3. El estudiante calcula la salida para el código B (figura 6.10) y podemos notar, que los resultados que obtuvo mostrados sobre la tabla le generan los mismos patrones de salida que los que obtuvo para el código B (figura 6.8).

ii. ¿qué es lo que están haciendo los códigos A y B? explica su salida

inicializa la posicion del peon, y mientras no haya una pieza delante o este en la esquina del tablero, se mueve

Fig. 5.8

```
7.      int Peon[7][7];
8.      int i=3;
9.      while(not(Peon[i][0] == Peon[0][0]  and  Peon[i-1][0] != NULL)){
10.              Peon[i][0] = Peon[i-1][0];
11.              if(Peon[i-1][0] == Peon[0][0]){
12.                  printf("la maquina ha ganado");
```

Fig. 5.9 código A

```
7.      int Peon[7][7];
8.      int i=3;
9.      while(not(Peon[i][0]==Peon[0][0])  or  not(Peon[i-1][0] != NULL)){
10.             Peon[i][0] = Peon[i-1][0];
11.             if(Peon[i-1][0] == Peon[0][0]){
```

Fig. 5.10 código B

2.4. La comparación que hace el estudiante **EF**, realizada mediante las salidas obtenidas en el código A y el código B, se hace presente en el reactivo *iii* (Figura 6.11). De esta manera, podemos deducir a partir de la respuesta que el estudiante género en iii (figura n), ambos códigos son equivalentes, dadas las salidas que obtuvo en ambos códigos.

iii. ¿Crees que es conveniente para la maquina el haber preferido utilizar alguno de los dos códigos? ¿Por qué?

Creo que es mas conveniente usar el codigo A, usa menos operadores

Fig. 5.11

3. Objetos de salida: el estudiante **EF** parece encontrar una relación de equivalencia entre las líneas 9 de ambos códigos, las expresiones son equivalentes:

$\boldsymbol{not}(Peon[i][0] == Peon == [i][0]\ \boldsymbol{and}\ Peon == [i-1][0]\,! = NULL)$ y $\boldsymbol{not}(Peon[i][0] == Peon == [i][0])\ \boldsymbol{or}\ \boldsymbol{not}(Peon == [i-1][0]\,! = NULL)$

Observaciones correspondientes a la primera ley.

Podemos concluir para esta sección que en los contextos que fueron abordados en general no se mostró gran complicación por parte de los estudiantes, exceptuando que, sí existieron casos aislados de estudiantes que no tenían las construcciones necesarias tales como conceptos de programación particularmente el uso de ciclos simples y anidados así como también el manejo fluido de los arreglos bidimensionales y finalmente el uso de los operadores lógicos, esto con el propósito de poder seguir con el resto de la actividad.

5.2.2 Acción Segunda Ley

Definición Acción Segunda Ley $\neg(\boldsymbol{p} \lor \boldsymbol{q})$

En esta segunda parte de las actividades se evalúa la segunda ley de De Morgan, por lo que estaremos evaluando prácticamente los mismos elementos que fueron vistos para la primera ley, es por ello que conservaremos la estructura nos enfocaremos en mostrar los ejemplos correspondientes. Los elementos de esta Acción son:

1. Objetos de entrada:
   1.1. Un par de proposiciones $(\boldsymbol{p}, \boldsymbol{q})$ y sus negaciones
2. Pasos/operaciones sobre los objetos:

En los siguientes puntos se hace una descripción de lo que se busca encontrar para la concepción Acción mostrando una relación detallada.

   2.1. Calcular la disyunción $\boldsymbol{p} \lor \boldsymbol{q}$
   2.2. Calcular la negación de la disyunción $\neg(\boldsymbol{p} \lor \boldsymbol{q})$
   2.3. Calcular la conjunción de las negaciones de los elementos $\neg\boldsymbol{p} \land \neg\boldsymbol{q}$, y si es necesario previamente calcular las negaciones $\neg p$, $\neg q$.
   2.4. Comparar $\neg(\boldsymbol{p} \lor \boldsymbol{q}), \neg\boldsymbol{p} \land \neg\boldsymbol{q}$
3. Objetos de salida: El elemento final de la concepción Acción es la relación entre los elementos de salida que obtuvo en 2.2 y 2.3, lo que propicia que se logre una reflexión sobre la equivalencia $\neg(p \lor q) = \neg p \land \neg q$, es decir que *la negación de una disyunción es equivalente a la conjunción de las negaciones*.

Contexto 1: Desigualdades

Para este contexto que corresponde a la segunda ley, hacemos una pequeña observación con respecto al primer ejercicio de las desigualdad, pues pudimos notar que casi la gran mayoría de los estudiantes presentaron dificultades al momento de identificar la solución, podemos sospechar que el motivo fue que la solución que se esperaba encontrar era el conjunto vacío por lo que por ese aspecto podemos argumentar que, refiriéndonos al conjunto vacío ya por si solo puede representar un dificultad para la mayoría de los estudiantes.

Presentamos las respuestas del estudiante **EF**, quien completó las actividades, podemos decir también que durante la sesión presento algunas dudas menores que tenían que ver con la interpretación de los reactivos y no con las construcciones previas.

1. Objetos de entrada:

Estudiante **EF** identificó en la primera expresión de las instrucciones $not(x \geq 5\ or\ x \leq 8)$, hizo una distinción entre las proposiciones p y q y el operador lógico. Por consiguiente, para él las proposiciones son las siguientes desigualdades: $p: x \geq 5$, $q: x \leq 8$, $\neg p: x < 5$, $\neg q: x > 8$.

2. Pasos/operaciones sobre los objetos:
   2.1. El estudiante **EF** calcula primero $\boldsymbol{p} \vee \boldsymbol{q}$, esto lo realiza dejando indicado arriba de la recta dos círculos que representan el intervalo abierto (Figura 6.12), los cuales corresponden a los valores de la expresión contenida dentro del paréntesis, es decir al conjunto vacío.

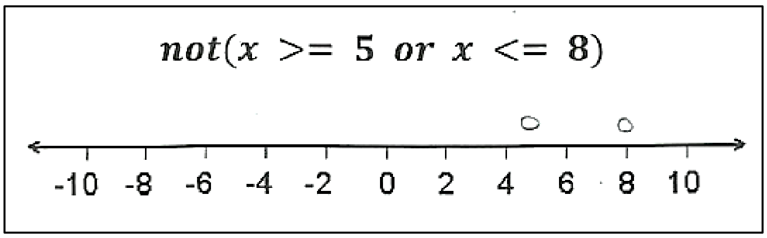


Fig. 5.12

   2.2. Para $\neg(\boldsymbol{p} \vee \boldsymbol{q})$, el estudiante señala los intervalos abiertos de $(\boldsymbol{p} \vee \boldsymbol{q})$.
   2.3. El estudiante **EF** calcula la conjunción $\neg\boldsymbol{p} \wedge \neg\boldsymbol{q}$ sobre la recta: expreso al igual que en la parte de arriba, dos círculos con el intervalo abierto (Figura 6.13), mismos que representan que para este problema el valor no tiene solución, aquí podemos creer que el estudiante se adelantó y de alguna manera empleo las operaciones en la mente pues no presenta ninguna evidencia escrita donde se haya hecho el cálculo, esto es debido a que su solución corresponde a la solución que es esperada.

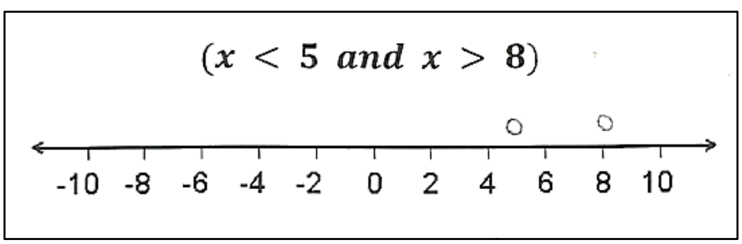


Fig. 5.13

   2.4. Comparar las proposiciones $\neg(\boldsymbol{p} \wedge \boldsymbol{q}) = \neg\boldsymbol{p} \vee \neg\boldsymbol{q}$, a partir de la pregunta que se plantea en el reactivo ii (Figura 6.14). El estudiante **EF** concluye que las expresiones son equivalentes, a través de la reflexión sobre sus respuestas del inciso *i.*

ii. Compara ambos resultados de la gráfica y determina qué relación encuentras entre ambos casos, anota tus observaciones.

Son iguales y no generan un conjunto solucion

Fig. 5.14

3. Objetos de salida: Es entonces que el estudiante **EF** concluye la relación de equivalencia correcta.

Contexto 2: Trabajo con el Applet.

De igual manera y como se mencionó, aquí el estudiante estará interactuando con el ambiente dinámico.

Presentamos las respuestas por el estudiante **ED**, quien completó las actividades, durante la sesión de trabajo pudimos percatarnos tenía algunas dudas las cuales estaban concentradas en la redacción de los reactivos, pues al parecer al estudiante le parecían un poco vagas.

1. Objetos de entrada:

El estudiante **ED** identifico las proposiciones contenidas en las columnas P y Q de la tabla (véase Figura 6.15), de igual manera como lo hizo en el ejercicio los ingreso en el applet.

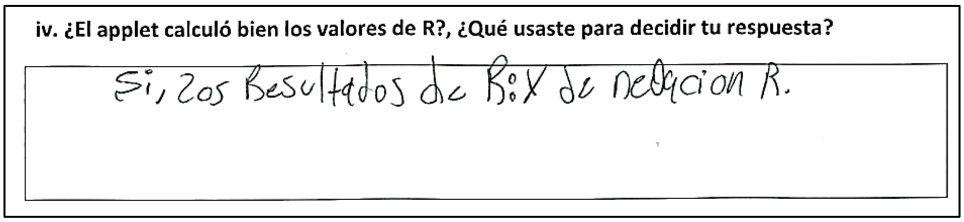
Caso	P	Q	R	$not(R)$
1	$x < 10$	$x > 20$	(X<10) or (X>20)	not(X<10) and not(X>20)
2	$x < 5$	$x > 9$	(X<5) or (X>9)	Not – and – Not
3	$x < -3$	$x > 12$	(X<3) or (X>12)	not – and – not
4	$x < 0$	$x > 9$	(X<0) or (X>9)	Not – and – Not

Fig. 5.15

2. Pasos/operaciones sobre los objetos:
   2.1. De igual manera que para la primera ley, la operación de calcular $(\boldsymbol{R} = \boldsymbol{P} \vee \boldsymbol{Q})$ fue procesada por el ambiente dinámico.
   2.2. Las operaciones de calcular $\neg(\boldsymbol{R})$ de igual manera es procesado internamente por el ambiente dinámico y es por parte del alumno comparar la expresión que se generó en $(\boldsymbol{R})$ con las disponibles para $\neg(\boldsymbol{R})$ de la lista desplegable de la parte derecha del applet, este contexto pretende agilizar el contexto de desigualdad visto anteriormente
   2.3. El estudiante **ED** calcula el operador and identificando y evaluando en la gráfica del lado derecho la opción correcta es decir; la disyunción y la conjunción de las negaciones

y lo representa en la tabla de las hojas de trabajo y las representaciones del ambiente dinámico.

2.4. El estudiante **ED** completa la tabla de una manera particular, es decir el individuo después de escribir la expresión en el primer renglón, este ya no escribe los valores para la negación de R, pero sí los operadores, podemos creer que para el individuo los operadores seguirán siendo admitidos para cualquier P, Q que la expresión contenga. Finalmente, el estudiante **ED** compara los elementos obtenidos del llenado de la tabla (Figura 6.16) y los resultados de la gráfica obtenidos por el applet.


iv. ¿El applet calculó bien los valores de R?, ¿Qué usaste para decidir tu respuesta?

Si, los Resultados de R: Y de negacion R.

Fig. 5.16

3. Objetos de salida:

Podemos ver en la tabla y en la respuesta a la pregunta *iv* que, el estudiante **EC** concluye que al aplicar la negación a un operador lógico, para el caso de una disyunción, se niegan las expresiones contenidas y se cambia del operador lógico por una conjunción. Por lo que el individuo parece prever que siempre la negación de una conjunción será disyunción de las proposiciones negadas.

Contexto 3: Ajedrez (código)

Tomaremos las respuestas proporcionadas por el estudiante **ED**, pues es el que menos presento complicaciones para este ejercicio en particular.

1. Objetos de entrada:

En este contexto los elementos de entrada son de igual manera las proposiciones que se encuentran dentro de bucles de repetición de los códigos C (figura 6.18) y D (figura 6.19). El estudiante **ED** necesariamente identifica las proposiciones p y q en la condición del bucle de repetición para poder ejecutarlo (línea 9 Figuras 6.18, 6.19), de esta manera obtiene que: $p: i < 3$ y $q: i > 6$.

2. Pasos/operaciones sobre los objetos:

   El estudiante **ED** calcula $(\boldsymbol{p} \vee \boldsymbol{q})$

2.1. y 2.2. En el código D, el estudiante **ED** identifica $\boldsymbol{p} \vee \boldsymbol{q}$ y el operador negación not, por lo que procede con las iteraciones del ciclo for y representa las iteraciones sobre la tabla, generando los resultados correctos (Figura 6.17).

	0	1	2	3	4	5	6	7
0								
1								
2								
3		1						
4		2						
5		3						
6		4						
7		5						

Jaque mate

Fig. 5.17

2.3. El estudiante **ED**, ejecuta el código D, obteniendo la misma salida que en el código C, las condiciones necesarias para poder realizar las iteraciones se encuentran en el cuerpo o condicional de los ciclos for.

```
int peonNegro[7][7];
int i;
for(i=3;not(i<3 or i>6);i++){
        peonNegro[i][1] = peonNegro[i+1][1];

        if(peonNegro[i][1] == peonNegro[7][1])
            printf("jaque mate");
      }
```

Fig. 5.18 código C

```
int peonNegro[7][7];
int i;
for(i=3;(i>=3 and i<=6);i++){
        peonNegro[i][1] = peonNegro[i+1][1];

        if(peonNegro[i][1] == peonNegro[7][1])
            printf("jaque mate");
    }
```

Fig. 5.19 código D

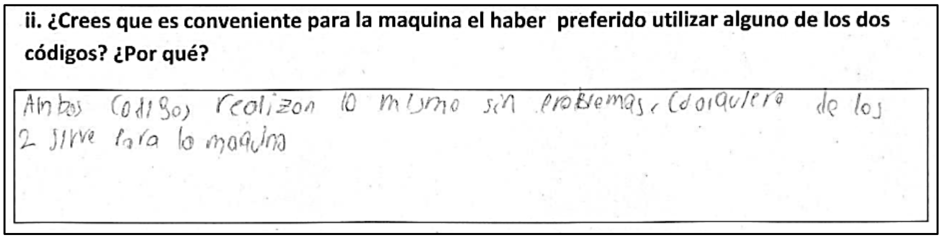
Esto lo podemos comprobar con la respuesta que proporciono el reactivo *i* (Figura 6.20)

i. ¿Cuál es el resultado que obtuviste con los códigos C y D?

Mismo Resultado, ultima iteración fue jaque mate.

Fig. 5.20

2.4. La comparación que hace el estudiante **ED**, la realiza mediante las salidas obtenidas en el código C y el código D se hace presente en el reactivo *i* (Figura 6.20) y *ii* (Figura 6.21). El estudiante encuentra una relación entre las tablas donde escribió las iteraciones y concluye que los códigos son equivalentes según su salida.


ii. ¿Crees que es conveniente para la maquina el haber preferido utilizar alguno de los dos códigos? ¿Por qué?

Ambos codigos realizan lo mismo sin problemas, cualquiera de los 2 sirve para la maquina

Fig. 5.21

3. Objetos de salida: podemos ver que el estudiante **ED** afirma que ambos códigos son equivalentes por medio de la respuesta que proporciono en el reactivo anterior.

Observaciones correspondientes a la segunda ley.

Para la segunda ley, podemos notar algunas inconsistencias en la mayoría de las respuestas de los estudiantes, las cuales contrastan con las dificultades encontradas en el caso de la primera ley. Particularmente en el contexto de las desigualdades, esto se puede rastrear desde dos posibles situaciones: la primera, puede ser debido a que los estudiantes no identifican al vacío como un conjunto; y la segunda, que a los estudiantes aún les falta trabajar la parte de los operadores lógicos.

En el contexto del ajedrez pudimos encontrar algunas dificultades en algunos de los participantes, estos presentaban complicaciones para representar los ciclos de repetición en el contexto de programación, incluso algunos no pudieron siquiera abordar los problemas. Particularmente los ciclos de repetición for causaron una dificultad mayor que los ciclos $while$ trabajados en la sección anterior, pero podemos percibir que el motivo fue que los ciclos de repetición presentados en esta sección son de carácter más abstracto que los de la primera sección.

El contexto 4: Tablas de verdad de ambas leyes

En este contexto se pretendió proporcionar una generalización de ambas leyes de De Morgan, con lo cual esto involucra un acercamiento al individuo a que trabaje con una notación a nivel lógico, las nociones que logro el individuo con anterioridad es que cada ley la trabajaba de manera individual, así de esta manera se pretende que el estudiante pueda encontrar una relación entre ambas leyes

En este ejercicio no se pudo lograr si es que tal generalización se puedo llevar a cabo, puesto que lo esperábamos es que el estudiante pudiera dar una clara expresión de la igualdad de las columnas (Figura 6.22), tan solo una respuesta proporcionada por el estudiante **EF** se acercó a lo que estipulábamos, pero se nos imposibilita proporcionar sólidos argumentos sin contar con más resultados.

El análisis a este contexto se muestra en la parte de abajo y es similar al que se ha estado trabajando en los otros contextos solo que este se aplica a ambas leyes.

1. Objetos de entrada:
   1.1. Las proposiciones $(\boldsymbol{p}, \boldsymbol{q})$, correspondientes a las columnas A y B de ambas tablas, así como su expresión en su forma de proposición molecular.

El estudiante **EF** identifica las entradas A, B, $A\ and\ B$, $A\ or\ B$, las cuales son las columnas de las tablas A y B, primeramente, resolviendo la tabla A, la cual corresponde a la primera ley es decir la negación de una conjunción, posteriormente hace lo mismo con la tabla B, segunda ley.

2. Pasos/operaciones sobre los objetos:
   2.1. El estudiante calcula la negación de la conjunción empleando el valor $A\ and\ B$ y $A\ or\ B$ el cual es proporcionado en la tabla; $\boldsymbol{p} \vee \boldsymbol{q}$, $\boldsymbol{p} \wedge \boldsymbol{q}$
   2.2. Calcular la negación de la disyunción y negación de la conjunción $\neg(\boldsymbol{p} \vee \boldsymbol{q})$, $\neg(\boldsymbol{p} \wedge \boldsymbol{q})$
   2.3. Calcular la conjunción de las negaciones de los elementos $\neg\boldsymbol{p} \wedge \neg\boldsymbol{q}$, y si es necesario previamente calcular las negaciones $\neg p$, $\neg q$.
   2.4. Comparar $\neg(\boldsymbol{p} \vee \boldsymbol{q})$ **,** $\neg\boldsymbol{p} \wedge \neg\boldsymbol{q}$,$\neg(\boldsymbol{p} \wedge \boldsymbol{q})$ **,** $\neg\boldsymbol{p} \vee \neg\boldsymbol{q}$ (las columnas de la tabla)
3. Objetos de salida: El elemento final de la concepción Acción es la relación entre los elementos de salida que obtuvo en 2.2 y 2.3 los cuales los puede comparar directamente al comparar ambas tablas, lo que propicia que se logre una reflexión sobre la equivalencia $\neg(p \vee q) = \neg p \wedge \neg q$, $\neg(p \wedge q) = \neg p \vee \neg q$ es decir que *la negación de una disyunción es equivalente a la conjunción de las negaciones, así como lo es que la negación de la conjunción es equivalente a la disyunción de sus proposiciones negadas*.

Tabla A

A	B	A and B	not (A and B)	not A and not B
Verdad	Verdad	Verdad	Falso	Falso
Verdad	Falso	Falso	verdad	Falso
Falso	Verdad	Falso	Verdad	Falso
Falso	Falso	Falso	Verdad	verdad

Tabla B

A	B	A or B	not A or not B	not (A or B)
Verdad	Verdad	Verdad	Falso	Falso
Verdad	Falso	Verdad	Verdad	Falso
Falso	Verdad	Verdad	Verdad	Falso
Falso	Falso	Falso	Verdad	Verdad

ii. ¿Qué relación pudiste encontrar entre las últimas dos columnas de la tabla A y la tabla B, encuentras alguna similitud?

algunas columnas son iguales

Fig. 5.22

5.2.3 Procesos para las leyes de De Morgan

Definición de Proceso para las leyes de De Morgan

En las actividades anteriores se favoreció la repetición de la Acción, con esto esperamos que se llegue a interiorizar como Proceso. Las concepciones de Proceso para cada una de las Leyes de De Morgan se puede caracterizar de tal manera que ahora el individuo no tiene la necesidad de ejecutar los pasos como: identificar los elementos de entrada, las proposiciones $(p\ , q)$, calcular la conjunción y la disyunción, calcular sus negaciones y posteriormente hacer la comparación de las expresiones con la finalidad de determinar si son equivalentes. Podemos decir que ahora el individuo puede omitir o imaginar los pasos para negar una conjunción o disyunción y determinar su valor de salida, porque su experiencia previa lo convenció de que siempre se cumple una equivalencia específica.

Proceso de la primera ley $\neg(\boldsymbol{p}\ \wedge \boldsymbol{q})$

El individuo obtiene la expresión de *negación de una conjunción* como una expresión que adquirió en Acción mediante la repetición, por tanto, puede relacionar el resultado de la expresión con su respectiva equivalencia sin tener que realizar las operaciones mencionadas, realizándolo de la siguiente manera:

$$\neg(\boldsymbol{p}\ \wedge \boldsymbol{q})\ \equiv\ \neg\boldsymbol{p}\ \vee \neg\boldsymbol{q}$$

Es aquí donde el estudiante ya piensa en el caso general y concluye que siempre el lado izquierdo es igual al derecho, por tanto, aplica el caso general y es en este momento que el individuo puede aplicar las leyes como fórmulas.

Proceso de la segunda ley $\neg(\boldsymbol{p} \lor \boldsymbol{q})$

De igual manera, el individuo identifica la expresión de *negación de una disyunción* como una expresión que adquirió en Acción mediante la repetición, entonces el individuo puede relacionar el resultado de la expresión con su respectiva equivalencia sin tener que realizar las operaciones, con lo que intuye que la solución es:

$$\neg(\boldsymbol{p} \lor \boldsymbol{q}) \equiv \neg\boldsymbol{p} \land \neg\boldsymbol{q}$$

Aquí el individuo ya no tiene la necesidad de comparar la equivalencia de la expresión, el individuo debido a su experiencia que adquirió en Acción, confía en su validez y concluye que siempre el lado izquierdo es igual al derecho y para cualquier valor de p y q. Es decir, lo visualiza como el caso general por tanto es en este momento que el individuo puede emplear las leyes como formulas.

Contexto Tabla de las dos leyes

En este primer ejercicio de la actividad dos, tiene como propósito determinar si los estudiantes superaron la concepción Acción que se trabajó durante toda la actividad uno, y ver si ahora pueden trabajar con Proceso. En la tabla no se tienen espacios añadidos donde se pida que se calculen los pasos, por lo que ahora se está sugiriendo la concepción Proceso.

Un estudiante que se encuentra a nivel Acción se vería en la necesidad de que emplear operaciones para corroborar que sus resultados son verdaderos.

El estudiante **EF** está implementando las leyes de De Morgan directamente sobre los reactivos de la tabla, esto nos da un indicio de que se está presentando la concepción Proceso. En las respuestas que proporciona podemos notar que hace uso directo de las leyes de De Morgan, por ejemplo, en el segundo renglón, para este contexto las expresiones de las desigualdades están en forma general y no para valores determinados dificultándosele el que quiera implementar las operaciones. Un individuo con concepción proceso utiliza el hecho de que sin importar que valor represente la variable, *la ley siempre se cumple*, por lo que esto no debería de generar un obstáculo para dar la expresión equivalente.

En la tabla de abajo (Figura 6.23) podemos percatarnos como es que el estudiante **EF** aplica el Proceso en la mayoría de los reactivos, es decir escribe directamente como quedaría la expresión después de aplicar las leyes, esto nos da un indicativo que el estudiante está avanzando con el desarrollo de la concepción Proceso.

A	B	$not(A\ and\ B)$	$not(A\ or\ B)$
$x < a$	$x >= b$	not(X<a and X≥b)	not(X<a or X≥b)
$x > b$	$x == b$	X≤b or X!=b	X≤b and X!=b
"Juan es cocinero"	*"pedro es policía"*	Juan no es cocinero or Pedro no es Policía	Juan no es cocinero And Pedro no es Policía
$\neg(R)$	$\neg(S)$	R or S	R and S
(R)	$\neg\neg(S)$	¬R or ¬S	¬R and ¬S
$a\,!= b$	$a\,!= c$	not(a!=b and a!=c)	not(a!=b or a!=c)

Fig. 5.23

Pasar de la concepción Acción a la concepción Proceso no es una tarea sencilla pues este conlleva un esfuerzo particular. A continuación, mostraremos un ejemplo del estudiante **EJ,** que no mostró avances en la concepción Proceso y se quedó solo a nivel Acción, esto lo pudimos confirmar pues su desempeño en los ejercicios anteriores demuestra que el individuo no ha podido desempeñarse bien con los conceptos requeridos.

A	B	$not(A\ and\ B)$	$not(A\ or\ B)$
$x < a$	$x >= b$	not(x<a and x>=b)	not(x<a and x>=b)
$x > b$	$x == b$	not(x>b and x==b)	not(x>b or x==b)
"Juan es cocinero"	*"pedro es policía"*	Juan no es cocinero y Pedro no es Policia	Juan no es cocinero o Pedro no es Policia
$\neg(R)$	$\neg(S)$	R and S	R or S
(R)	$\neg\neg(S)$	¬R and ¬S	¬R or ¬S
$a\,!= b$	$a\,!= c$	(a==b and a==c)	not(a!=b or a!=c)

Fig. 5.24

Observemos en la tabla de arriba (Figura 6.24), como fue que el estudiante fue resolviendo los espacios de la tabla, podemos decir sustituyendo los valores en los primeros dos renglones, justo después y para el resto de los reactivos, empezó a implementar algunas propiedades de las leyes de De Morgan como la doble negación, pero no lo pudo concretar exitosamente pues como podemos ver, no incluyó el cambio del operador lógico. Por esto, podemos decir que el estudiante aún no ha adquirido la concepción Proceso y tampoco concepción Acción pues existe una debilidad notoria en sus construcciones previas, es decir aun no logra interiorizar las leyes de De Morgan como una fórmula, así como las propiedades que la conforman.

Contexto de Código para demostrar las leyes de De Morgan

En esta sección solo contamos con muy pocos resultados pues tan solo unos pocos estudiantes llegaron a esta etapa y dentro de ellos solo un par de ellos pudieron contestar correctamente lo que se requería.

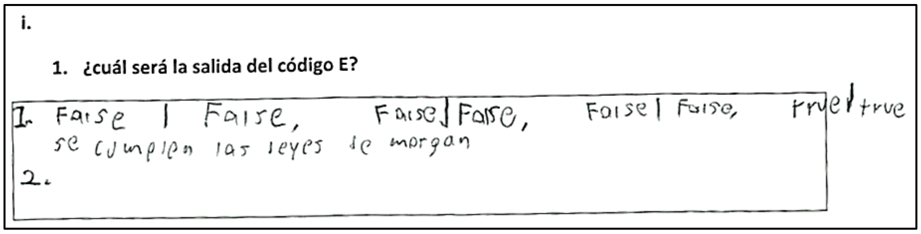
Tomaremos las respuestas proporcionadas por el estudiante **ER,** podemos ver que no resulto de gran complicación al encontrar la salida esperada, observamos en la figura n que el individuo hace algunas anotaciones de apoyo.

Mostraremos los reactivos para el inciso *i*.

```
5. int main()
6. {
7.        bool A[] = {true, false};
8.        bool B[] = {true, false};
9.        bool C[4] = {};
10.       bool D[4] = {};
11.       bool iguales = true;
12.       int contador=0;
13.
14.           for(int x=0;x<2;x++){
15.               for(int y=0;y<2;y++){
16.                 C[contador] = not(A[x] or B[y]);
17.                 D[contador] = not(A[x]) and not(B[y]);
18.                 cout<<C[contador]<<"|"<<D[contador]<<endl;
19.                    contador = contador + 1;
20.                 }                              contador = 1
21.           }
22.           for(int x=0;x<contador;x++){
23.                 if(C[x] != D[x])
24.                       iguales = false;         iguales = true
25.                       break;
26.           }
27.           if(iguales)
28.                       cout << "se cumple la ley de Morgan" <<endl;
29.           else
30.                       cout << "No se cumple la ley de Morgan" <<endl;
31.       }
```

Fig. 5.25

Para las expresiones en las líneas 16 y 17 del código anterior (Figura 6.25), deducimos que el individuo puede imaginarse la salida de las operaciones por lo que es buen indicio de que cuenta con la concepción Proceso de los ciclos anidados, además por la correcta implementación que logro de la línea 23 podemos visualizar que también cuenta el Proceso de las leyes pues pudo imaginar la salida, misma que es el resultado de la tabla de verdad para la segunda ley, pues ésta es la que determina si ambas salidas darán como resultado la equivalencia.

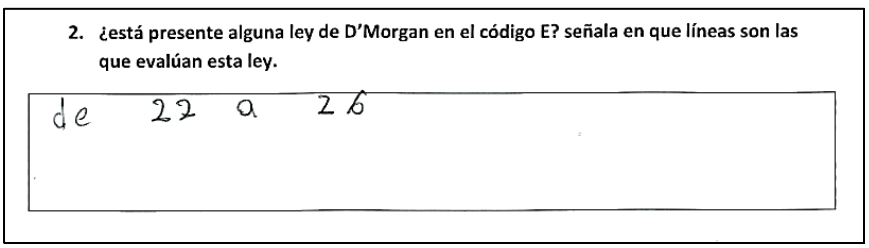

i.

1. ¿cuál será la salida del código E?

1. False | False, False | False, False | False, true | true
se cumplen las leyes de morgan
2.

Fig. 5.26

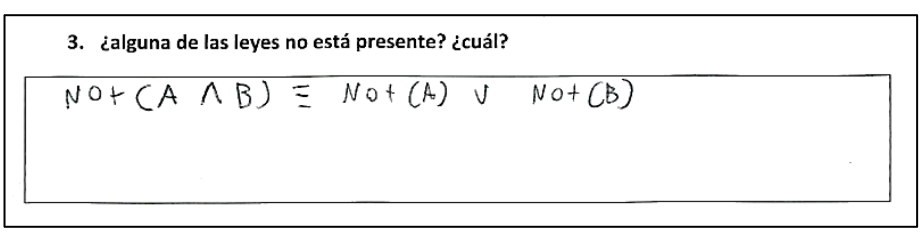
En las líneas 18, 28 del código (figura 6.25) se solicita como salida en consola un (*cout*) la tabla de verdad que generaba el mismo código y el mensaje final formaron parte del resultado del código. El hecho que el individuo escribiese la tabla mostrada en ejercicio anterior (Figura 6.26), no es señal de que no haya implementado la concepción de Proceso, sin embargo. Si la línea 18 no hubiera sido parte del código y él estudiante se viera en la necesidad de escribir cada salida que genera el código, es decir la tabla de verdad, entonces podríamos sospechar que el individuo no ha adquirido la concepción Proceso y en su lugar está todavía en la etapa de Acción.


2. ¿está presente alguna ley de D'Morgan en el código E? señala en que líneas son las que evalúan esta ley.

de 22 a 26

Fig. 5.27

En el ejercicio anterior (Figura 6.27) correspondiente al reactivo 2, podemos ver como el estudiante **ER** es capaz de distinguir las líneas del código que están evaluando la expresión, las cuales comprenden al ciclo de repetición for completo. Por otro lado, podemos suponer que debido a la redacción del reactivo 2, el individuo genera ese tipo de respuesta, sin embargo para el reactivo 3, que mostramos en la parte de abajo (Figura 6.28), vemos que responde correctamente; esto nos demuestra que el estudiante es capaz de identificar en el código (figura n) que falta una expresión correspondiente a la otra ley. Es decir, el estudiante fue capaz de identificar la expresión sin tener que hacer operaciones para corroborarlo, por lo tanto cuenta con la concepción Proceso.

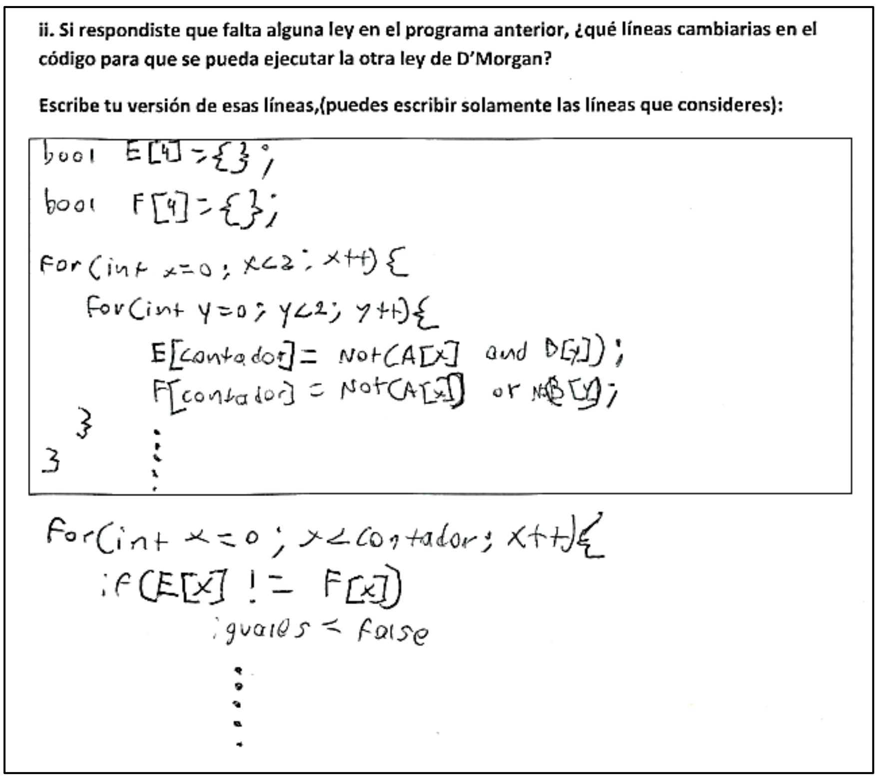

3. ¿alguna de las leyes no está presente? ¿cuál?

$Not(A \wedge B) \equiv Not(A) \vee Not(B)$

Fig. 5.28

Ahora mostraremos las respuestas del inciso *ii,* generadas por el mismo estudiante **ER**, dando una continuación de la sección completa.

Aquí lo que pide el reactivo es que se logre completar la sección del código que es necesaria para poder ejecutar la segunda ley de una manera correcta. Podemos ver en el siguiente reactivo (figura 6.29), el estudiante generó un código, la correcta estructura de éste implica que el estudiante no tiene dificultades con la concepción proceso de los ciclos de repetición anidados, según Cetin (2015). Además, podemos ver que **ER** maneja las leyes como formulas pues no tuvo que generar la salida del código para comprobar que ésta seria correcta. En este caso, el estudiante logro responder de forma correcta, pues solamente cambio las expresiones de las líneas 16 y 17 correspondientes a las leyes; es decir, el estudiante confió en que si ejecutara el código éste arrojaría una respuesta verdadera sin tener que realizar los pasos de la ejecución.


ii. Si respondiste que falta alguna ley en el programa anterior, ¿qué líneas cambiarias en el código para que se pueda ejecutar la otra ley de D'Morgan?

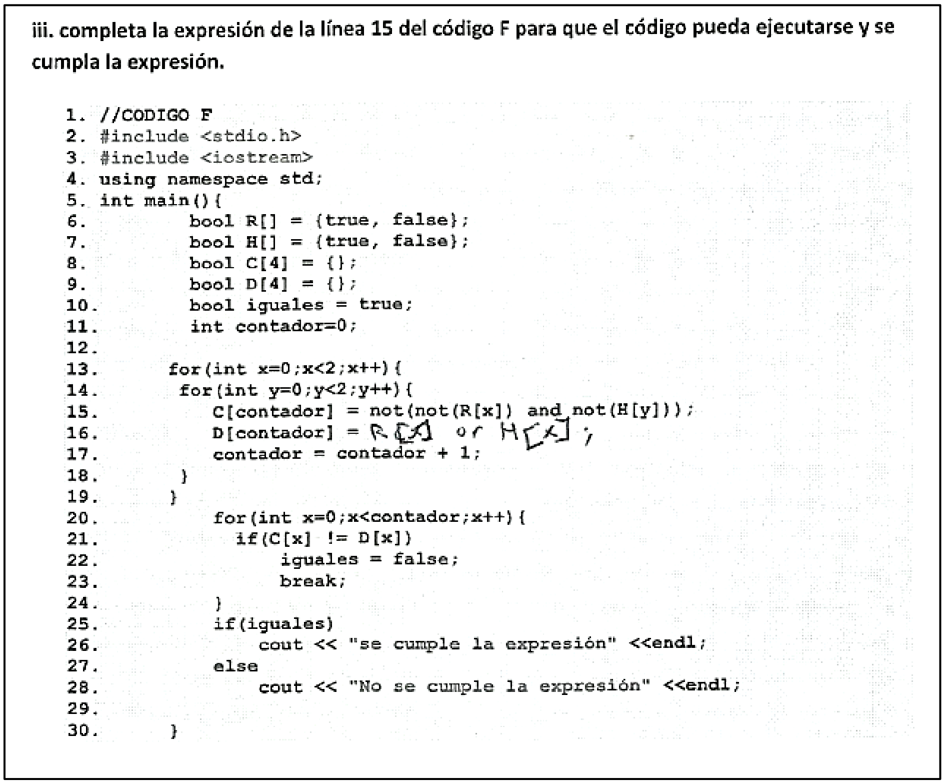
Escribe tu versión de esas líneas,(puedes escribir solamente las líneas que consideres):

```
bool E[4]={};
bool F[4]={};

for(int x=0; x<2; x++){
    for(int y=0; y<2; y++){
        E[contador]= Not(A[x] and B[y]);
        F[contador] = Not(A[x]) or Not(B[y]);
    }
}       :

for(int x=0; x<contador; x++){
    if(E[x] != F[x])
        iguales = false
        :
```

Fig. 5.29

En el inciso iii (figura 6.30), podemos ver el estudiante muestra un claro avance con la concepción Proceso, de esta manera el individuo ya tiene una distinción entre las leyes de De Morgan y aplica directamente la ley sobre las líneas 16 y 17 en el código (figura 6.30), para completar la equivalencia en el código y su salida sea verdadera, de igual manera podemos ver que el individuo no realiza ninguna operación, lo cual es un indicativo de Proceso.

El estudiante es capaz de identificar que en el código anterior solo se estaba presentando una ley, por lo que el individuo es capaz de identificar la otra expresión como ley, es interesante además como este incorpora el ciclo que hace la comparación entre ambos arreglos, pues este es el que se encarga de validar que la ley se cumpla.



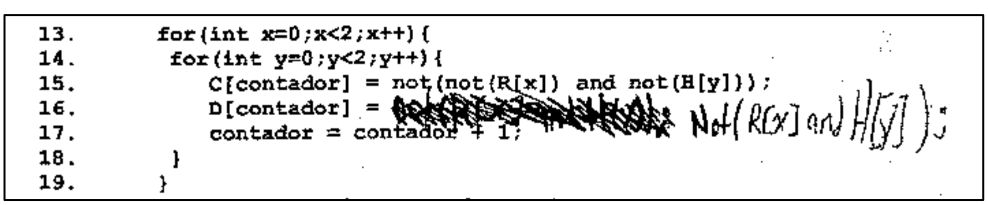
iii. completa la expresión de la línea 15 del código F para que el código pueda ejecutarse y se cumpla la expresión.

```
1.  //CODIGO F
2.  #include <stdio.h>
3.  #include <iostream>
4.  using namespace std;
5.  int main(){
6.          bool R[] = {true, false};
7.          bool H[] = {true, false};
8.          bool C[4] = {};
9.          bool D[4] = {};
10.         bool iguales = true;
11.         int contador=0;
12.
13.       for(int x=0;x<2;x++){
14.        for(int y=0;y<2;y++){
15.           C[contador] = not(not(R[x]) and not(H[y]));
16.           D[contador] = R[x]  or  H[x] ;
17.           contador = contador + 1;
18.        }
19.       }
20.           for(int x=0;x<contador;x++){
21.             if(C[x] != D[x])
22.                 iguales = false;
23.                 break;
24.           }
25.           if(iguales)
26.                cout << "se cumple la expresión" <<endl;
27.           else
28.                cout << "No se cumple la expresión" <<endl;
29.
30.       }
```

Fig. 5.30

El estudiante puede aplicar las leyes directamente si tener que realizar ningún tipo de pasos, por lo que nuevamente es un identificativo de que está empleando el Proceso de los ciclos anidados, otra forma de poder visualizarlo es que dentro de su respuesta no se vio en la necesidad de hacer ningún calculo esto también nos dice que el estudiante tiene el Proceso de las leyes de De Morgan.

Sin embargo, algunos estudiantes no pudieron responder este reactivo de manera satisfactoria, por lo que sospechamos que aún no han interiorizado bien los conceptos ya sean en los contextos de lógica o de programación, a continuación mostraremos algunos ejemplos.



```
13.       for(int x=0;x<2;x++){
14.        for(int y=0;y<2;y++){
15.           C[contador] = not(not(R[x]) and not(H[y]));
16.           D[contador] =  Not( R[x] and H[y] );
17.           contador = contador + 1;
18.        }
19.       }
```

Fig. 5.31

En reactivo anterior (Figura 6.31) podemos ver que el individuo aún no ha adquirido la concepción Proceso, es decir solo puede manejar los conceptos a nivel Acción donde son requeridos los pasos para encontrar una solución. Parece que trabaja sólo nivel de manipular las expresiones, sin actuar sobre los objetos representados, se limita a cancelar algunos símbolos de negación.

```
for(int x=0;x<2;x++){
 for(int y=0;y<2;y++){
    C[contador] = not(not(R[x]) and not(H[y]));
    D[contador] = not(not(R[x])) or not(not(H[y]));
    contador = contador + 1;
 }
}
```

Fig. 5.32

Por otro lado hacemos una distinción en la clase de respuestas que son proporcionadas por alguno de los estudiantes, en la sección de código que mostramos arriba (figura 6.32) podemos hacer una observación que si bien la respuesta es correcta, puede implicar debilidades en las construcciones previas, por ejemplo el caso de la doble negación.

5.2.4 Procesos generalizados

La generalización de las leyes de De Morgan

En este apartado, se espera que los estudiantes cuenten con los procesos de las leyes para dar la solución de este ejercicio, aplicar las leyes de De Morgan para el caso generalizado. Por el tipo de estudiantes a los que van dirigidas las actividades, no esperamos que proporcionen soluciones rigurosas en el sentido matemático, pero sí se espera que tengan la idea intuitiva de que las leyes de De Morgan son válidas para *n* proposiciones.

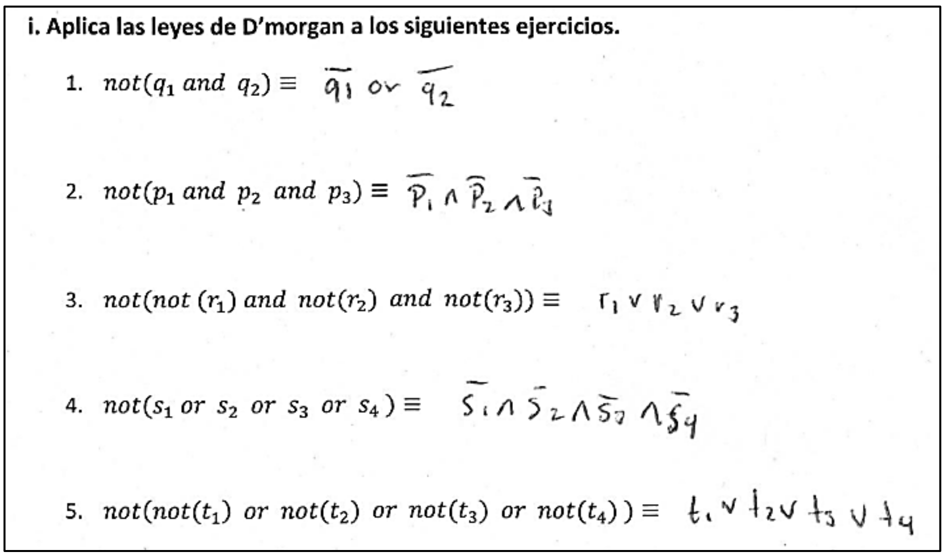
Tomaremos las respuestas proporcionadas por el estudiante **ED**, en el reactivo que se muestra en la parte de abajo (Figura 6.33) el estudiante comienza empleando las leyes a las expresiones que se muestran, pero ahora como formulas.

i. Aplica las leyes de D'morgan a los siguientes ejercicios.

1. $not(q_1\ and\ q_2) \equiv$ not(q1) or not(q2)
2. $not(p_1\ and\ p_2\ and\ p_3) \equiv$ not(q1) or not(q2) or not(q3)
3. $not(not\ (r_1)\ and\ not(r_2)\ and\ not(r_3)) \equiv$ not(not(r1)) or not(not(r2)) or not(not(r3))
4. $not(s_1\ or\ s_2\ or\ s_3\ or\ s_4) \equiv$ not(s1) and not(s2) and not(s3) and not(s4
5. $not(not(t_1)\ or\ not(t_2)\ or\ not(t_3)\ or\ not(t_4)) \equiv$ not(not(t1)) and not(not(t2)) and not(not(t3)) and not(not(t4))

Fig. 5.33

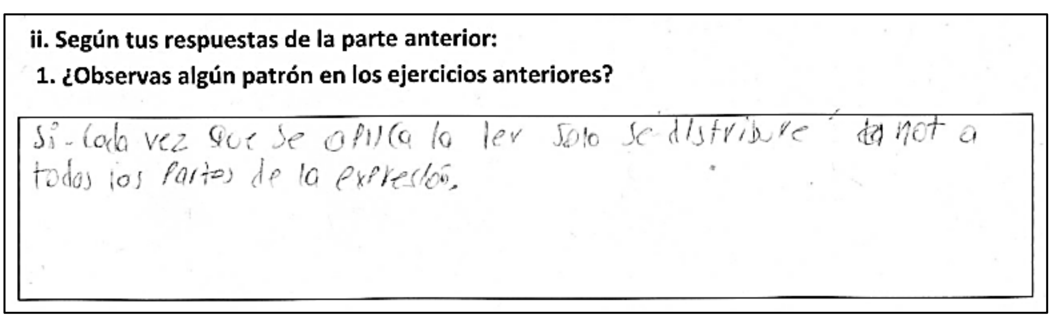
Podemos observar por otro lado un ejemplo en el cual no se logra la concepción Proceso, el reactivo mostrado en la parte de abajo (figura 6.34) se ve el caso de fallo en los incisos 2, 4 y 5. Esto es debido a que el individuo no ha interiorizado del todo las leyes como formulas, el error recurrente mostrado tiene que ver con el cambio del operador al aplicar la negación, sin embargo se puede ver que el individuo hace uso la notación que es empleada para circuitos y compuertas lógicas. Por tanto podemos decir que en este caso el estudiante no ha logrado la interiorización.


i. Aplica las leyes de D'morgan a los siguientes ejercicios.

1. $not(q_1\ and\ q_2) \equiv$ $\overline{q_1}$ or $\overline{q_2}$
2. $not(p_1\ and\ p_2\ and\ p_3) \equiv$ $\overline{p_1} \wedge \overline{p_2} \wedge \overline{p_3}$
3. $not(not\ (r_1)\ and\ not(r_2)\ and\ not(r_3)) \equiv$ $r_1 \vee r_2 \vee r_3$
4. $not(s_1\ or\ s_2\ or\ s_3\ or\ s_4) \equiv$ $\overline{s_1} \wedge \overline{s_2} \wedge \overline{s_3} \wedge \overline{s_4}$
5. $not(not(t_1)\ or\ not(t_2)\ or\ not(t_3)\ or\ not(t_4)) \equiv$ $t_1 \vee t_2 \vee t_3 \vee t_4$

Fig. 5.34

Podemos observar varios indicadores del Proceso buscado, dada la respuesta que generó en el reactivo de abajo (figura 6.35) que además es continuación del ejercicio anterior, podemos distinguir que siempre que las expresiones las distingue como leyes que siempre que se encuentra con una negación esta se distribuye a las expresiones que están dentro.


ii. Según tus respuestas de la parte anterior:

1. ¿Observas algún patrón en los ejercicios anteriores?

Si - cada vez que se aplica la ley solo se distribuye da not a todas las partes de la expresión.

Fig. 5.35

Para el reactivo 2 mostrado en la parte de abajo (Figura 6.36), podemos ver en el estudiante **ED**, proporciona un argumento dentro de sus respuestas que las tambas de verdad de ambas expresiones son distintas, si bien no contestamos que los cálculos los realizo en una hoja externa a las hojas de trabajo, sí podemos ver en las otras respuestas que género, argumentos tales como que si se aplicara la propiedad asociativa a las expresiones, entonces obtendría resultados distintos. Podemos concluir que el estudiante puede visualizar más allá de las operaciones y es entonces que esta imaginando las respuestas que obtendría, por lo que podemos asegurar que es un caso evidente de la concepción Proceso.

2. Compara las expresiones $not(p_1\ and\ (p_2\ or\ p_3))$ y $not((p_1\ and\ p_2)\ or\ p_3))$.
¿Son iguales o distintas?

distintos
tabla de verdad distinto
Asociativo mal aplicada
dan diferentes valores por los operadores logicos.

Fig. 5.36

Para el inciso *iii*, mostrado en la figura de abajo (Figura 6.37), las preguntas 1 y 2 tienen como propósito la generalización de ambas leyes vistos desde la concepción Proceso.

iii.

1. Propón expresiones para la negación de la disyunción y la conjunción para el caso de n **premisas de acuerdo a las leyes de D'Morgan.**

2. Explica verbalmente que significan esas expresiones.

1 Sea $a_1, a_2, \ldots, a_n$ proposiciones logicas.

2 not(a_1) or not(a_2) or ... or not(a_n)
es igual a:

3 not(a_1 and a_2) or not(a_3 and a_4) or ... or not(a_{n-1} and a_n)
es igual a:

4 not ((a_1 and a_2) and (a_3 and a_4)) or ... or not ((a_{n-3} and a_{n-2}) and (a_{n-1} and a_n))
⋮

5 not(a_1 and a_2 and ... and a_n)

6 not(a_1) and not(a_2) and ... and not(a_n)
es igual a:

7 not(a_1 or a_2) and not(a_3 or a_4) and ... and not (a_{n-1} or a_n)
⋮

8 not(a_1 or a_2 or ... or a_n)

Fig. 5.37

Las respuestas mostradas a continuación corresponden al estudiante **EF**, primeramente podemos notar que la demostración que proporciono, empezó a asignar parejas de proposiciones a partir de la línea tres, es decir $not(a_1 \; and \; a_2) \; or \dots \; or \; not(a_{n-1} \; and \; a_n)$ y fue aplicando las leyes consecutivamente hasta obtener la forma resultante: $not(a_1 and \; a_2 \; and \dots and \; a_n)$ mostrada en la línea ocho, como una pequeña observación y para fines prácticos no daremos mucha importancia a la forma en como expreso la solución, debido a que comenzó demostrando la parte derecha de la equivalencia de la primera ley $\neg(p \wedge q) \equiv \neg p \vee \neg q$. Podemos ver también que el individuo expresa de una manera generalizada las proposiciones etiquetándolas inicialmente con $a_{1,} a_2, \dots, a_n$. Siendo a_n, cualquier cantidad de proposiciones en las cuales se cumpliría la ley asociativa de las leyes de De Morgan.

Podemos ver claramente la intervención de la concepción Proceso, esto debido a que el individuo está trabajando con proposiciones a nivel abstracto y además está empleando el uso de las leyes de De Morgan como formulas, sin embargo también notamos que el individuo está aplicando Procesos sobre Procesos, esto nos lleva a concluir que se está realizando una encapsulación de Proceso, lo que da como resultado la concepción Objeto.

6 Conclusiones

A continuación, mostraremos algunos resultados que se obtuvieron a partir del análisis de las actividades desde el punto de vista del marco teórico, esto con el propósito de comprobar la existencia de las concepciones mentales que pretendíamos buscar (Acción, Proceso y Objeto).

6.1 Sobre las concepciones buscadas

6.1.1 Resultados de la concepción Acción

Pudimos ver que algunos de los individuos pudieron generar los pasos necesarios para la concepción Acción; sin embargo, esto no siempre se cumplió. Notamos que algunos individuos daban saltos en los pasos que propusimos para Acción. Esto lo pudimos notar en las respuestas que proporcionaban; pero no lo pudimos comprobar pues las únicas evidencias con las que se contaron fueron las actividades de la puesta en escena y las respuestas escritas en ellas; sin embargo, sí pudimos ver que los individuos utilizaban hojas alternas para elaborar algunos borradores, a los cuales no pudimos acceder.

Las acciones vienen dadas por las construcciones previas, en la puesta en escena pudimos notar la ausencia de algunas construcciones previas, esto después de aplicar las actividades de la puesta en escena, por lo que se prevé que en una segunda interacción se puedan agregar.

6.1.2 Resultados de la concepción Proceso

Algunos estudiantes lograron contestar las actividades demostrando que contaban con la concepción Proceso; es decir, sin necesidad de emplear la serie de pasos que exige una Acción; sin embargo, para el desarrollo de esta sección pocos estudiantes lograron concluir las actividades. Pudimos encontrar algunos casos aislados en donde el individuo asimilaba bien el concepto y podía aplicarlo hacia cualquier entorno de trabajo. En cuyos casos podemos decir que el individuo comprende las leyes de De Morgan como un Objeto. Por otra parte, para el resto de los participantes y de los que lograron contestar al menos parte de esta sección, se hizo una distinción clara de que algunos de los individuos podían interiorizar las leyes de De Morgan como fórmulas y tenían una clara visión de lo que estas expresan; sin embargo, y es un dato interesante, algunos de ellos no podían manejarlas del todo bien para los diferentes contextos. Pareciera que al moverse de un contexto a otro, el individuo tiene la noción de que está trabajando en otro tipo de problema y que además es aislado del resto.

Algunas de las fallas comunes y que esperábamos encontrar fueron las deficiencias en las construcciones previas, en particular el manejo de los ciclos anidados. Este aspecto es comprensible pues sabíamos que el individuo tenía que realizar un trabajo extra para coordinar ambos Procesos, el de los ciclos anidados y el concepto de las leyes de De Morgan.

6.1.3 *Resultados de la concepción Objeto*

Algunos estudiantes mostraron ejemplos interesantes que demuestran que pasaron de la concepción Proceso a la concepción Objeto, esto se logra cuando el individuo es capaz de realizar Procesos sobre Procesos, esta situación la pudimos ver en la sección donde se trabajaba con la generalización de las leyes de De Morgan.

6.1.4 *Procesos avanzados*

Los procesos avanzados se caracterizan no por ser procesos sobre procesos, más bien por aplicar el proceso a un elemento que previamente requiere un cambio de variable, es decir que para llegar de un estado a otro en la solución de un problema este tiene que pasar por una transformación primeramente, tiene que ser tratado mediante una notación especial o cambiar alguna componente de la expresión misma, para que una vez teniendo esto se pueda aplicar el proceso simple a un elemento de manera implícitamente, el resultado se obtiene una vez descomponiendo la expresión.

6.1.5 *Construcciones previas*

Las nociones constructivistas afirman que el conocimiento parte de las nociones que el estudiante ya posea y que a partir de nuevas herramientas empieza a construir nuevo conocimiento. Sin embargo, vimos necesario una sección que nos garantizara que los individuos contaran con las herramientas necesarias para poder darle el apropiado seguimiento en el resto de las actividades. Esta es nombrada *actividad cero*, esta actividad se evaluó desde el punto de vista del marco teórico por lo que de igual manera fueron diseñadas para promover las concepciones de Acción y Proceso, pero no se incorporó al análisis como tal, pues solo consistió en actividades diagnóstico.

6.1.6 *Complejidad de los problemas de programación*

Uno de los objetivos del trabajo de tesis fue implementar ejercicios en donde se pudieran resaltar aplicaciones hacia contextos reales, en particular en el contexto de programación. En estos contextos fue fácil ver que algunos de los estudiantes no contaban con las construcciones necesarias o previas para poder resolver los ejercicios; añadimos que se esperaban este tipo de complicaciones, pues pudimos anticipar que para lograr un desarrollo en los ejercicios se debía de contar con construcciones previas; es decir que tuvieran como mínimo las construcciones de Acción y Proceso de los ciclos de programación. En la mayoría de los estudiantes pudimos observar que pudieron identificar las variables y condicionales; sin embargo, fue en los ciclos de repetición simples y los anidados, así como en los arreglos bidimensionales donde existieron las complicaciones. Algunos de los detalles más sobresalientes fueron el manejo de los índices en los arreglos. Otras complicaciones notorias y observaciones interesantes fueron que si bien dentro de las actividades, cuando estaban trabajando otro contexto, algunos de los estudiantes no presentaron dificultad, sí lo hicieron en el contexto de programación, por lo que podemos sospechar que esos individuos no han podido interiorizar el concepto matemático y trasladarlo hacia otro contexto.

6.2 Reflexiones finales

6.2.1 Descomposición genética preliminar

Mediante el análisis que se ha realizado podemos concluir que las construcciones de Acción y Proceso son necesarias para construir el concepto de las leyes de De Morgan, si bien las actividades fueron diseñadas para favorecer esas construcciones y para ese tipo particular de estudiantes, pudimos observar que algunos estudiantes contaban con construcciones previas, con lo que pudimos notar que los estudiantes operaban Procesos sobre Procesos trabajándolos en una coordinación por lo que la concepción Objeto forma parte de la construcción del concepto de las leyes de De Morgan, el cual fue un resultado que no previmos cuando desarrollamos las actividades esto lo pudimos observar pues dentro de las actividades en una sección se exigía que los estudiantes contaran con construcciones previas como lo fue el caso de la sección de los códigos de programación, que por su nivel de complejidad estos ya exigían la concepción Proceso, por lo que esto nos llevó a concluir que las leyes de De Morgan se pueden observar de igual manera como una concepción Objeto, aun así no es fácil y no es un tema trivial en el aprendizaje de este tema en particular, por lo que se requiere un esfuerzo.

Una manera razonable de comprender el proceso que está detrás y que es lo que genera la forma recomendable para desarrollar el aprendizaje del concepto matemático de las leyes seria comenzando con una concepción Acción de las Leyes de De Morgan. Esta comienza cuando el individuo puede tomar un par de proposiciones p y q para posteriormente operar sobre ellas mediante un conectivo lógico ya sea de disyunción o conjunción; de igual manera puede tomar dos proposiciones las cuales pueden ser las mismas que ya tenía, p y q. Estas están unidas por un conectivo lógico, siempre y cuando este sea distinto al que fue tomado en la primera expresión, y opera sobre estas dos expresiones de manera aislada, lo cual es necesario pues como hemos visto las leyes de De Morgan primeramente se deben de trabajar individualmente para posteriormente unirlas. La manera de hacerlo es por medio de una intervención que consta de preguntas que van guiando al individuo para que este pueda generar una reflexión. Este tipo de ejercicios se va plantearon de manera progresiva para que el individuo pudiera generar una idea general de lo que estaba sucediendo; esto lo llevaría aque pueda interiorizar bien el concepto e imaginar los pasos o las operaciones sin tener que efectuarlos, lo que lo conducirá a una concepción de Proceso.

A continuación, mostraremos dos caminos posibles que pueden llegar a presentarse en el desarrollo de una descomposición genética, si bien ambos caminos culminan con la generalización de las leyes de De Morgan, presentan algunas variantes para concebirlas después de la primera etapa de interiorización.

En ambos caminos se desarrolla la comprensión del concepto desde Acciones a Procesos simples y luego a Procesos avanzados. Comenzando con las Acciones, éstas son propuestas como una manera eficaz de lograr las concepciones Proceso, pues la interiorización pudiera ser más accesible para los estudiantes que la coordinación de Procesos.

Los Procesos simples se logran cuando ya se ha adquirido la habilidad de reconocer a las expresiones como Fórmulas, el individuo puede operar con ellas en diversos contextos y coordinando estos con los Procesos de los conectivos reconoce las expresiones como leyes que pueden operar sobre cualquier cantidad de proposiciones, mostrando esta habilidad decimos que el individuo ha logrado la generalización de las leyes de De Morgan.

Primer camino: coordinación de Procesos generales

En este camino se propone que las leyes de De Morgan se estudian de manera aislada para llegar a su forma generalizada, esto a partir de la interiorización de Acciones específicas. El estudiante comienza trabajando con proposiciones y los conectivos lógicos, como *elementos de entrada*. Se trabaja sobre ejercicios, en distintos contextos, en los que se efectúan operaciones específicas sobre estos para que finalmente comparen sus resultados y establezcan una igualdad.

Se trabaja la misma comparación en distintos contextos con la finalidad de que se logre la reflexión sobre la repetición y así interiorizar las Acciones. El mecanismo de interiorización parece requerir que, durante la repetición de las Acciones, los estudiantes reflexionen sobre las propiedades recurrentes, sea cual sea el contexto que se esté manipulando, lo cual los conduce a las expresiones formales interpretadas como leyes; una para cada conectivo. De esta manera, los Procesos de las leyes de De Morgan están dados por la interiorización que consiguieron a partir de las Acciones iniciales, logrando utilizar las leyes como formulas, en diversos contextos. Las concepciones de las leyes como Proceso pueden ser coordinadas con otros procesos para la generalización de las leyes de De Morgan.

Mientras que los procesos *simples* de las leyes se usan en problemas relativamente sencillos como el caso de la expresión $\neg(p \wedge q)$ que consta solamente dos proposiciones y un conectivo lógico, los procesos generales se forman de expresiones de mayor extensión, al contar con un numero de proposiciones y conectivos indefinido y a su vez estos pueden estar alternados con el operador negación.

El papel que desempeña la coordinación de los Procesos generalizados de las leyes figura en que el individuo puede resolver ejercicios donde están involucradas ambas leyes, esto requiere no solo la presencia de ambos Procesos, sino que se trabajen en conjunto y no de mañerea aislada para aplicar una ley dentro de la otra al negar una proposición que contenga varios conectivos.

Primer camino:

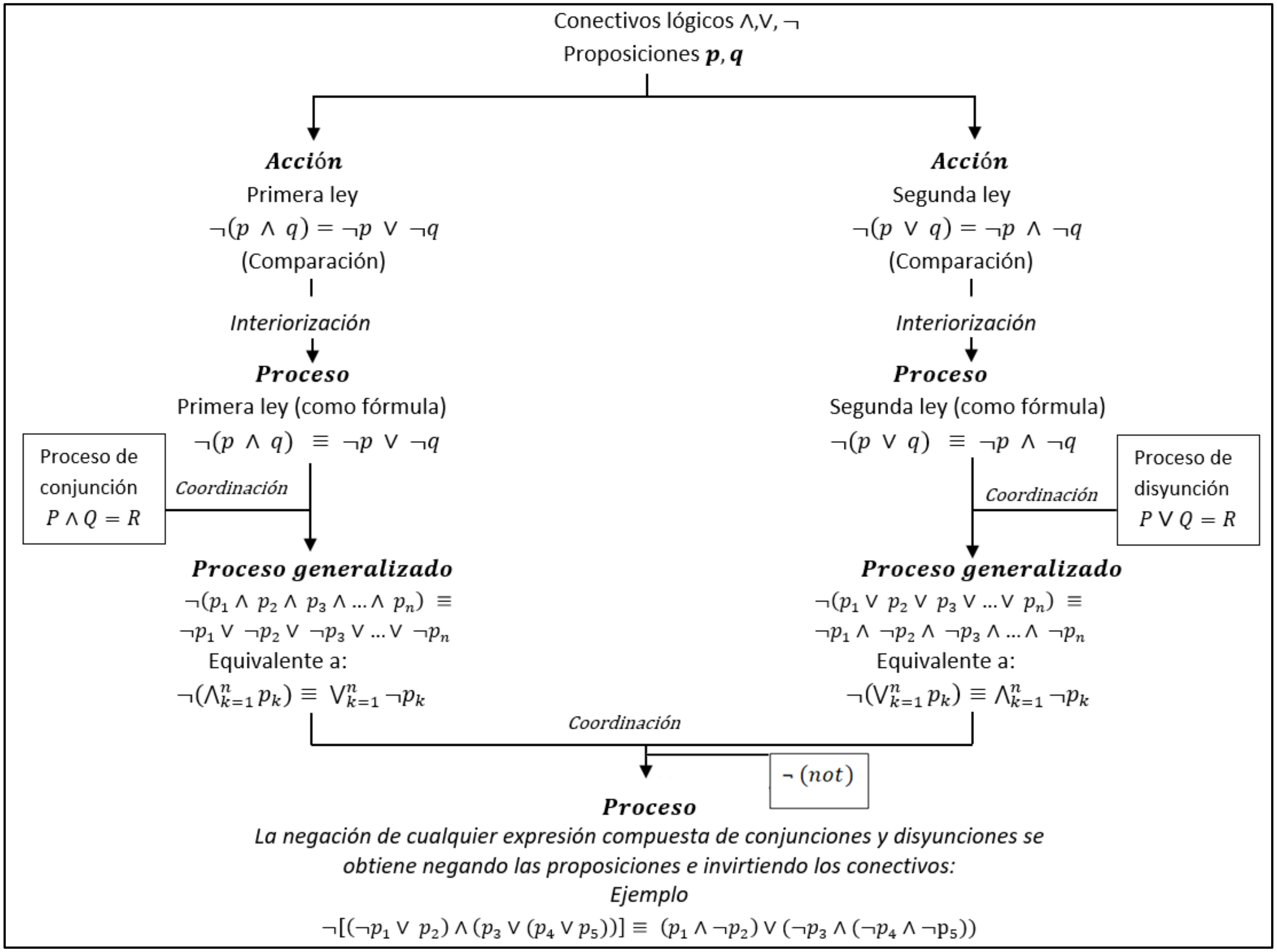


Fig. 6.38

Segundo camino: Generalización de Proceso coordinado.

Este camino propone la interiorización de Acciones a los Procesos simples de las leyes, exactamente como lo realiza el primer camino. La diferencia principal es que el estudiante, contando con suficiente experiencia sobre los Proceso de operadores lógicos, podría coordinar los Procesos de ambas leyes, de manera que genere un nuevo Proceso. Para esta Coordinación, el individuo asimila una ley dentro de la otra, para negar por ejemplo la disyunción de conjunciones.

La generalización de las leyes de De Morgan a partir del Proceso anterior, se logra con la coordinación de los Procesos de los conectivos y el proceso simple de las dos leyes. Este camino supone que el individuo ya cuenta con las nociones suficientes como para manipular las leyes de De Morgan, tanto en sus diferentes contextos, así como en su forma generalizada para un indeterminado número de conectivos lógicos, de la misma manera que en el primer camino.

Segundo camino:

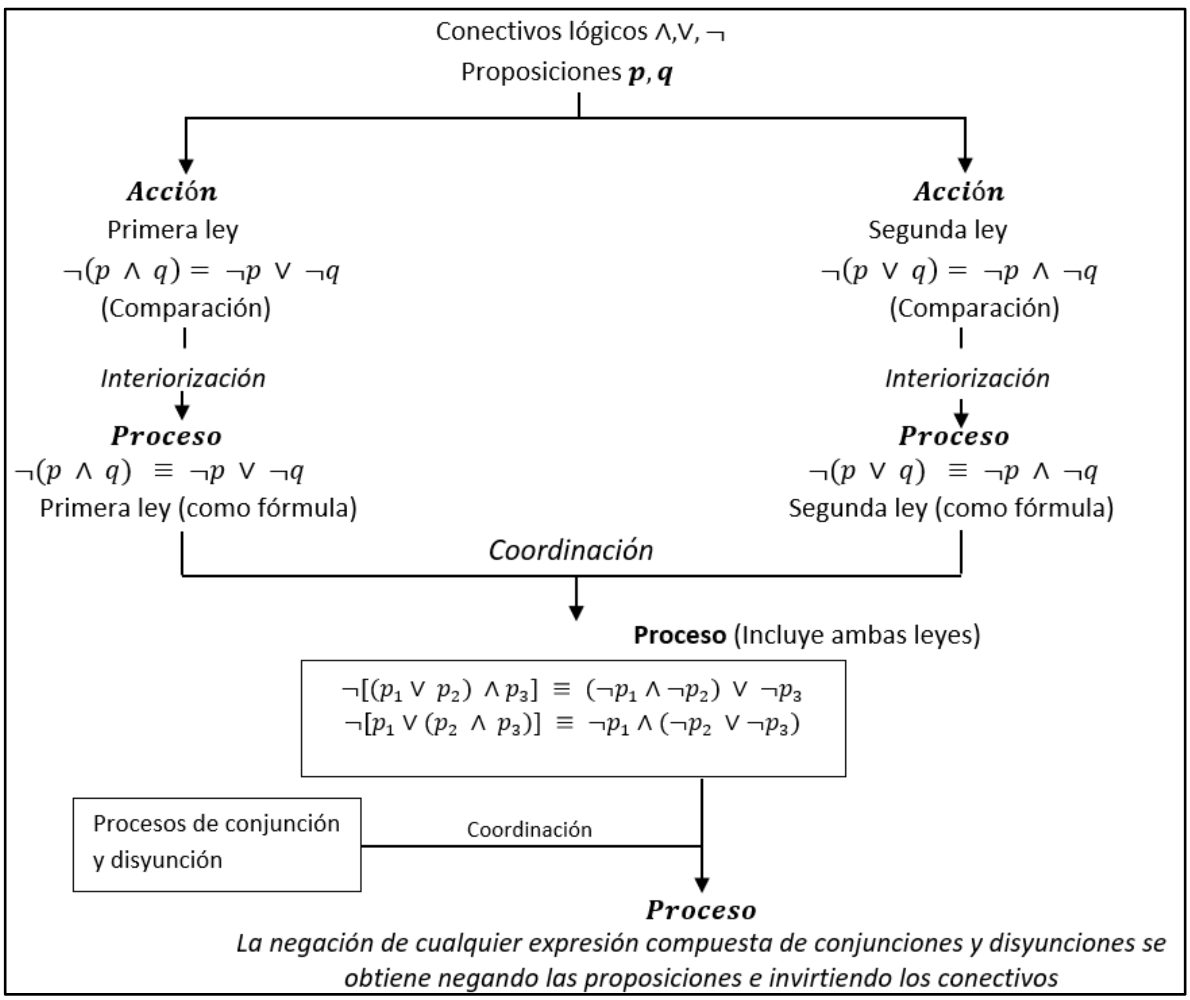


Fig. 6.39

En ambas propuestas es necesario distinguir entre emplear las leyes ciegamente y usarlas como fórmulas asociadas a Procesos, con esto queremos hacer referencia a que si el estudiante es capaz de resolver un problema a partir de la comprensión de lo que está haciendo y no solamente trate de sustituir valores. Esto es especialmente importante si es necesario moverse de contexto, por ejemplo, al de programación y el de la optimización de condicionales, no se trataría solamente de un ejercicio de sustitución, pues es necesaria una comprensión del concepto matemático.

En los procesos avanzados, se trabajan cuando el individuo además de coordinar Procesos, emplea en un esfuerzo extra, este se puede lograr de dos maneras, la primera es que el individuo tiene un buen manejo en las construcciones previas y en las operaciones lógicas como para expresar directamente las operaciones sin emplear pasos intermedios, en este caso podríamos decir que el individuo ya está dando signos de contar con la concepción de Objeto, es decir este puede visualizar las leyes de De Morgan como Objeto. La segunda manera es que el individuo aun no puede visualizar el resultado que obtendrá de una expresión al emplear las leyes, con lo que tiene que recurrir a el cambio de variable para poder efectuar directamente los pasos, es decir la forma natural de las leyes de De Morgan.

6.2.2 Versión general de la descomposición genética preliminar

Como hemos explicado anteriormente, el resultado obtenido de dos caminos viables para la descomposición genética surgió a partir del análisis y del tipo de resultados en los que estábamos enfocados. Sin embargo, las características compartidas de las coordinaciones observadas nos llevaron a idear una forma generalizada de la descomposición genética, la cual puede expresarse en términos simples. Este nuevo camino nos permite explorar algunos contextos dentro de la lógica de primer orden, en donde se involucre la asociación y el desarrollo de los operadores lógicos y la lógica proposicional, esto quiere decir que a partir del esquema sugerido es posible extenderlo a otros temas particulares dentro de los temas de la lógica Matemática.

De esta manera, la versión general de la descomposición genética (Fig. 6.40) se caracteriza por una coordinación de cinco procesos: las dos leyes como procesos simples y los tres operadores lógicos. Resaltamos que, aunque tal coordinación es teóricamente posible, ésta podría ser poco accesible para estudiantes como los observados. La complejidad de implementar secuencias de enseñanza con uno u otro camino sería uno de los problemas abiertos planteados a partir de nuestro trabajo.

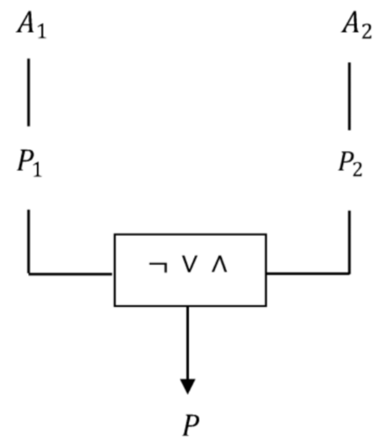


Fig. 6.40 Camino Generalizado

6.2.3 Implicaciones didácticas

Como hemos mencionado las leyes de De Morgan no es un tema trivial, aunque se pudiera mirar como un par de leyes las cuales no presentan mucha dificultad en donde solo hay que manipular un operador lógico. Esto lo pudimos comprobar al momento de observar las actividades, pues corroboramos que el individuo al trabajar en distintos contextos tales como ejercicios de rutina o en contextos de aplicación, su desempeño fue diferente y de ahí notamos las dificultades particulares.

Para la enseñanza del concepto de las leyes de De Morgan en el salón de clases se recomienda el trabajo con situaciones que hagan reflexionar a los estudiantes esto se puede lograr mediante la abstracción reflexiva por medio de ejemplos aislados, es decir actividades que proporcionen la construcción del concepto. Se pudiera comenzar en primera instancia por la comprensión de los operadores lógicos y su relación con los conjuntos; posteriormente se podría exponer parte de las actividades que desarrollamos para la puesta en escena extendiendo más los temas y agregando más situaciones que pudieran surgir en diferentes contextos relacionados con la formación que se recibe. Los argumentos que proporcionamos los basamos en las observaciones que se analizaron, pues durante la revisión de las actividades nos dimos cuenta de que en los ejercicios que correspondían al llenado de las tablas no se encontraron grandes problemas pues la mayoría de los estudiantes resolvieron esas secciones sin ninguna dificultad; sin embargo sí encontramos detalles cuando los estudiantes tenían que resolver ejercicios o analizar códigos de programación, donde se requiere de la coordinación de Procesos (Cetin 2015), pues estos ya requerían de reflexión.

Sobre todo las construcciones mentales juegan el papel fundamental en esta sección, las actividades deben de comenzar por la repetición de o la mecanización sobre elementos esto para favorecer la interiorización, no se puede garantizar que se tengan resultados óptimos si es que se comienza con la concepción Proceso pues como lo menciona la teoría, el trabajar a nivel

Proceso es más difícil pues interviene el concepto de la coordinación, además como lo hemos mencionado estas actividades que empleamos son de carácter de evaluación por lo que no nos proporcionó resultados a nivel de enseñanza, como los que se pueden obtener en un estudio de las concepciones empleadas al nivel de enseñanza en clase, para ello se puede tratar de seguir la descomposición genética que se planteó como una guía para la enseñanza.

En mi opinión considero que existe un salto entre los conceptos matemáticos y los conceptos que se aprenden en programación independientemente del lenguaje que se esté empleando; es decir y representándolo en un ejemplo concreto, ocurre como el salto que existe entre los conceptos que se aprenden en algebra lineal, tales como las operaciones y soluciones de matrices y sus propiedades. Al momento de representarlas en término de arreglos y ciclos de repetición en un lenguaje de programación existe un primer desencuentro, pues se descontextualiza de momento lo que ya se traía de conocimiento, y lleva tiempo ponerlo nuevamente en contexto. Análogamente nos encontramos con los conceptos de los operadores lógicos que son tan recurrentes como lo son los ciclos y los arreglos en programación. Una propuesta para atender estos planteamientos sería que existiera un elemento intermediario o bien podríamos representarlo como una especie de puente que conecte ambos esquemas. Este puente podría surgir en los cursos formales ofrecidos en matemáticas si existiera una unión o relación directa con las aplicaciones reales que se asignan en las prácticas de programación, y que estas no solo se quedaran a nivel algorítmico. Estas dificultades las podemos definir a partir del marco teórico, pues estas están asociadas a la necesidad de trabajo mental que es requerido, así como la debida coordinación de procesos que son necesarios para las prácticas en programación como hemos mencionado en capítulos anteriores.

6.2.4 Sobre el alcance de los objetivos

Al inicio de este trabajo se habían planteado los objetivos generales, algunos de los más relevantes fueron la elaboración de actividades de enseñanza y la intervención de medios de tecnología que ayudarían a una mejor compresión del concepto a tratar sin embargo el mismo marco teórico nos llevó a desarrollar un trabajo más cargado hacia la investigación que hacia la intervención didáctica.

Una enseñanza bajo el marco teórico es posible cuando se cuentan con elementos suficientes para poder llevarla a cabo, el elemento más determinante es contar con una descomposición genética, la cual representó una desventaja en nuestro trabajo pues no se contó con algún trabajo previo en el que pudiéramos apoyarnos y que contara con las características que mencionamos en secciones anteriores.

A través de este trabajo se lograron cubrir los objetivos específicos que planteamos en un comienzo, los cuales fueron el desarrollo de instrumentos de evaluación, correspondientes a las actividades de las cuales pudimos extraer información y resultados detallados sobre si las concepciones que propusimos se estaban llevando a cabo, mismos que utilizamos para poder proponer una primera versión de la descomposición genética la cual llamamos preliminar finalmente con esos resultados a disposición nos facilitó la tarea de poder ofrecer recomendaciones de enseñanza correspondientes al tema, mismos que pueden ser empleados en un salón de clase.

Una vez que los objetivos específicos fueron puntualizados y trabajados cubriendo punto por punto, se nos facilita en gran medida atender el objetivo principal de este trabajo, el cual fue como es que se lleva a cabo el aprendizaje y la construcción del concepto de las leyes de De Morgan por medio de las construcciones mentales y que son enfocadas particularmente a cierto tipo de estudiantes en donde el trabajo con el concepto mismo es frecuentemente recurrido. Por lo que el desarrollo de la descomposición genética fue una de las partes principales de este trabajo de tal manera, los dos caminos presentados de la construcción mental que proponemos, junto con las recomendaciones para su enseñanza, constituyen nuestra respuesta para aclarar el aprendizaje de las leyes de De Morgan en las carreras de Ciencias de la Computación

6.2.5 Sobre la problemática general

Uno de las características que siempre son latentes dentro del área de la enseñanza de la matemática es que siempre se ronda bajo en una problemática que pareciera no concluir, por lo tanto fue debido al trabajo realizado donde presentamos una problemática que nos percatamos y que corresponde a los conceptos de la logia, particularmente en la comprensión de los conectivos lógicos como conjunción, disyunción, pudimos ver que una negación o una negación doble es más fácil de comprender que una disyunción o una conjunción. Estos no fueron los únicos temas en donde existían complicaciones, pudimos notar también que existen algunas confusiones en el área de las desigualdades particularmente cuando se están manejando intervalos abiertos o cerrados.

6.2.6 Sobre la utilidad de estos resultados para futuros trabajos

En estos resultados se muestra una descomposición genética preliminar, por lo que se podría utilizar de punto de partida para un trabajo posterior en otra iteración al ciclo de investigación con el fin de lograr una descomposición genética final, así mismo parte de los resultados y de los análisis elaborados en los resultados de los estudiantes se pueden aprovechar para fines de un tipo de estudio sonde se abarquen temas relacionados con la lógica, pues las leyes de De Morgan forman parte solo de algunas propiedades de la lógica, por lo que algunos temas se pueden extender y unirlos con los temas del algebra de la teoría de conjuntos así mismo se podría desarrollar un estudio completo sobre la lógica proposicional basándose en los resultados previos ya realizados.

6.2.7 Sobre problemas abiertos

Durante la sesión de la puesta en escena y el análisis realizado, pudimos observar algunos objetos de interés que no contemplamos en nuestro trabajo y que se fueron saliendo, por lo que consideramos que valdría la pena señalarlos para su estudio independiente a este trabajo.

Un primer problema es el estudio de los arreglos dimensionales y bidimensionales en alumnos que cursan materias que están relacionadas con programación, esto debido a que en nuestro análisis surgieron ciertas dificultades referentes a las prácticas de programación, consideramos que valdría la pena desarrollar un estudio sobre como relacionan los estudiantes los temas vistos en las clases correspondientes al algebra lineal o algebra de matrices y como lo relacionan con los lenguajes de programación.

Otro de los posibles problemas abiertos de estudio es la teoría de conjuntos el cual es un tema muy poco concurrente y que además no se ha hecho un estudio cognitivo bajo el marco teórico de APOE, algunas dificultades es que pudimos notar dificultades referente al significado del conjunto vacío, pues observamos que muchos estudiantes no distinguían el vacío como un conjunto, seria de interés investigar como los estudiantes relacionan la teoría de conjuntos con las nociones de la lógica para poder extender el contenido a propiedades particulares de la lógica booleana.

Referencias

Arrieche M. (2006). Un estudio exploratorio sobre el aprendizaje de nociones conjuntistas por maestros en formación. Acta Scientiae, 8(2), 11-18.

Arnon, I., Cottrill, J., Dubinsky, E., Oktaç, A., Roa Fuentes, S., Trigueros, M. & Weller, K. (2014). APOS Theory: A framework for research and curriculum development in mathematics education. doi:10.1007/978-1-4614-7966-6. New York: Springer.

Cetin, I. (2015). Students' Understanding of Loops and Nested Loops in Computer Programming: An APOS Theory Perspective. Canadian Journal of Science, Mathematics and Technology Education, 15(2), 155-170. doi: 10.1080/14926156.2015.1014075

Deitel,H & Deitel, P .(2008). Instrucciones de control. Como Programar en C++ (A. Vidal, Trad.). Mexico: Pearson Education

Gamboa, M. (2013). Construcción cognitiva de la raíz cuadrada una mirada desde la teoría APOE (Tesis de maestría no publicada). Pontificia Universidad Católica De Valparaíso, Chile.

McCauley, R., Grissom, S., Fitzgerald, S. & Murphy, L. (2015). Teaching and learning recursive programming: a review of the research literature. Computer Science Education, 25(1), 37-66. doi: 10.1080/08993408.2015.1033205

Macbeth, G., Razumiejczyk, E., Crivello, M. C., Bolzán, C., Pereyra Girardi, C. I., & Campitelli, G. (2014). Mental models for the negation of conjunctions and disjunctions. Europe's Journal of Psychology, 10(1), 135-149.

Macbeth,G., Razumiejczyk, E., Crivello, M.C. (2014) Efecto de la distorcion introspectiva sobre los errores de superficie en el razonamiento lógico. Boletin de Psicologia, No 111, (Julio 2014), pp.25-43.

McCauley, R., Grissom, S., Fitzgerald, S. & Murphy, L. (2015). Teaching and learning recursive programming: a review of the research literature. Computer Science Education, 25(1), 37-66. doi: 10.1080/08993408.2015.1033205

Rodríguez E. & Rodríguez I. (2005). Notas para el curso para programación en seudocódigo. Universidad de Sonora México.

Richard J. (2005). Matemáticas Discretas (Marcia A. Trad.). Mexico: Pearson Education

Salgado, H. & Trigueros, M. (2009). Conteo: una propuesta didáctica y su análisis. Educación Matemática, 21 (1), 91-117.

Vest F. (1981). College Students' Comprehension of Conjunction and Disjunction. Journal for Research in Mathematics Education, Vol. 12, No. 3 (May, 1981), pp. 212-219.

Anexos

Actividades de las Leyes de De Morgan

Actividad 0.

Sección 1.- abre el applet que está en el siguiente link http://www.mat.uson.mx/~teoriadeconjuntos/ y genera cuatro desigualdades y empieza ingresarlas en la parte izquierda de la tabla de abajo, una vez le encuentres la solución sobre la gráfica en el applet, ahora encuentra el valor que es su contrario y escríbelo en la parte de enseguida de la tabla

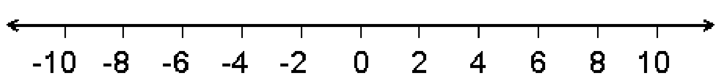
Desigualdades generadas	Su valor negado

Sección 2.-

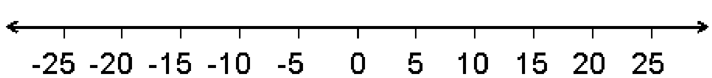
i. En la desigualdad $x \leq 8$, que valores está tomando x, según los siguientes incisos, marca la desigualdad más conveniente y escribe tus observaciones y por qué elegiste esa opción.

$$x < 8 \text{ or } x = 8 \qquad\qquad x < 8 \text{ and } x = 8$$

ii. Ahora si tomamos a $P: x \leq 8$ como expresión, qué región obtendríamos con la expresión $not(P)$, represéntalo en la gráfica.



iii. En la desigualdad $-5 < x$, qué valores tendríamos en la gráfica si tomamos todos los que no pertenecen a ese intervalo, señálalo en la gráfica y argumenta tu respuesta.



Sección 3.

En esta sección se presentaran unos ejercicios de programación, particularmente de ciclos de repetición y manejo índices.

i. cuál es la salida de los siguientes códigos

```
//CODIGO 1
#include <stdio.h>
#include <iostream>
using namespace std;
int main(){
    int x=0;
    for(int i = 1; i < 15 ; i++) {
         x=2*i;
        printf("\n%d",x);
    }
}
```

```
//CODIGO 2
#include <stdio.h>
#include <iostream>
using namespace std;
int main(){
    int n,i,suma;
    n=9;
    suma =0;

          i=1;
          while(i<n or i==n){
              suma = suma + i;
              i=i+1;
          }
          printf("la suma es %d",suma);
      }
```

Sección 4.

i. A continuación escribe el resultado que se obtiene al emplear el operador lógico negación, sobre cada expresión.

$$\boldsymbol{not(x < 6) =}$$

$$\boldsymbol{not(not(x \geq 9)) \ =}$$

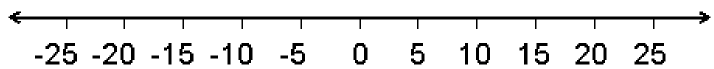
Actividad 1.

Sección 1.

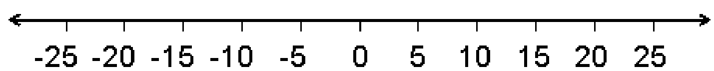
a) Primer caso

i. Representa las siguientes expresiones en su gráfica correspondiente.

$$(x < 10 \ or \ x > 20)$$



$$not(x >= 10 \ \ and \ \ x <= 20)$$



ii. Compara ambas gráficas y escribe que relación encontraste entre ellas.

Accede al enlace para responder la siguiente parte: [**http://www.mat.uson.mx/~teoriadeconjuntos/**](http://www.mat.uson.mx/~teoriadeconjuntos/)

iii. A continuación selecciona el caso P∧Q del lado izquierdo del applet e Ingresa la siguiente lista de valores que aparecen en la tabla de la columna P y la columna Q en las casillas del applet que acabas de acceder y completa la tabla para los valores que creas convenientes.

Caso	P	Q	R	$not(R)$
1	$x > 10$	$x \leq 20$		
2	$x > 5$	$x \leq 9$		
3	$x > -3$	$x \leq 12$		
4	$x \geq 0$	$x \leq 9$		

iv. ¿Qué observaste en las gráficas que obtuviste al ingresar los valores, te resulto alguna complicación?

v. ¿El applet calculó bien los valores de R?, ¿Qué usaste para decidir tu respuesta?

vi. ¿Cómo escribirías en términos de P y Q las expresiones de R y $not(R)$?

caso	R	$not(R)$
1		
2		
3		
4		

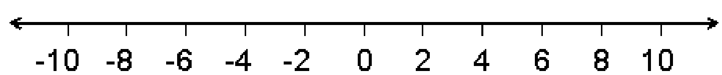
vii. ¿Hay algo que siempre se cumpla al ingresar los valores en la tabla?

viii. ¿La expresión $not(P \wedge Q) = notP \wedge notQ$ es posible el poder emplearla?, pruébalo en el applet y después escribe tus observaciones

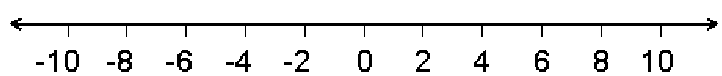
b) Segundo caso

i. Representa las siguientes expresiones en su gráfica correspondiente.

$$(x < 5 \ and \ x > 8)$$



$$not(x >= 5 \ or \ x <= 8)$$



ii. Compara ambos resultados de la gráfica y determina qué relación encuentras entre ambos casos, anota tus observaciones.

Accede al siguiente así como lo hiciste en el inciso a) del ejercicio anterior link: http://www.mat.uson.mx/~teoriadeconjuntos/

iii. A continuación selecciona el caso (P∨Q) del lado izquierdo del applet e Ingresa la siguiente lista de valores que aparecen en la tabla de la columna P y Q en las casillas del applet que acabas de acceder y completa la tabla para los valores que creas convenientes.

Caso	P	Q	R	$not(R)$
1	$x < 10$	$x > 20$		
2	$x < 5$	$x > 9$		
3	$x < -3$	$x > 12$		
4	$x < 0$	$x > 9$		

iv. ¿El applet calculó bien los valores de R?, ¿Qué usaste para decidir tu respuesta?

v. ¿Cómo escribirías en términos de P y Q las expresiones de R y $not(R)$?

caso	R	$not(R)$
1		
2		
3		
4		

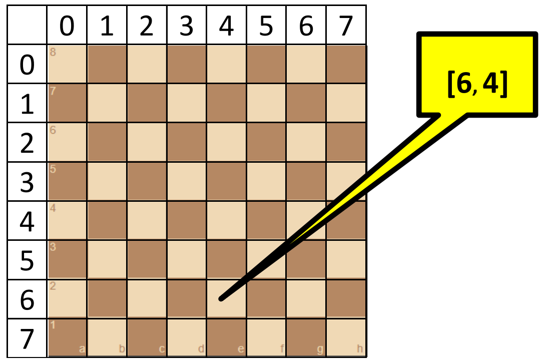
vi. ¿Hay algo que siempre se cumpla al ingresar los valores en la tabla?

vii. ¿La expresión $\boldsymbol{not(P \vee Q) = not(P) \vee not(Q)}$ es posible el poder emplearla?, pruébalo en el applet y después escribe tus observaciones

viii. ¿Qué relación encuentras entre los resultados del inciso a) y los resultados que obtuviste en el inciso b)?

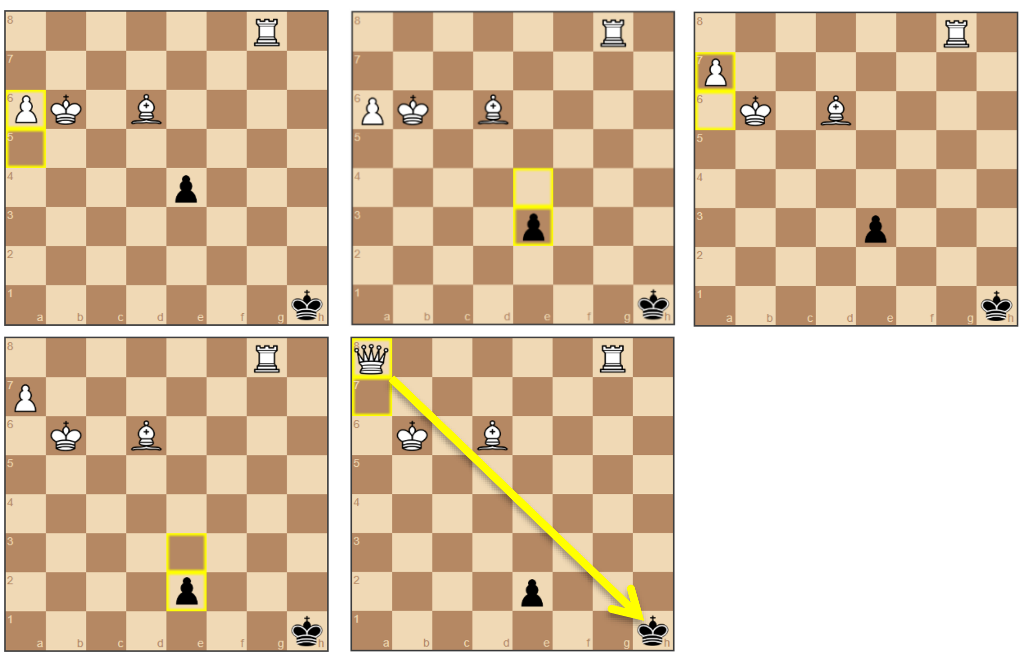
Sección 2.

En esta sección estaremos contemplando un entorno distinto, el juego de ajedrez, y para poder llevar a cabo esta sección daremos una breve explicación sobre la notación que estaremos utilizando, así pues la notación que puede leer la computadora es diferente a la que se usa comúnmente a la notación estándar del ajedrez, la notación de una computadora está representada por [filas][columnas]. En las coordenadas del tablero la lectura comienza desde el valor cero hasta el siete, precisamente como está representado en el siguiente diagrama, entonces podemos localizar fácilmente la posición que corresponde a (4,4) en el diagrama.



a) primer caso

En el siguiente diagrama se está llevando a cabo una partida de ajedrez, humano vs computadora siendo la maquina con piezas blancas y el humano con piezas negras.
El jugador con piezas negras no parece tener muchas opciones para poder mover, así que solo espera a que la maquina cometa algún error que le pueda dar ventaja para poder llegar a la primera fila para coronar, sin embargo, la maquina también efectúa su mejor movimiento.



Los código A y B están implementados en el lenguaje C++ y muestran una forma en como la computadora encontró la manera para poder ganar la partida, la maquina ejecuta el primer movimiento.

i. Ejecuta los códigos A y B, usa el diagrama para ayudarte y ve marcando sobre él las iteraciones que creas conveniente.

```
//CODIGO A
#include <stdio.h>
#include <iostream>
using namespace std;
int main()    //Código A
{
    int Peon[7][7];
    int i=3;
    while(not(Peon[i][0] == Peon[0][0]  and  Peon[i-1][0] != NULL)){
              Peon[i][0] = Peon[i-1][0];
              if(Peon[i-1][0] == Peon[0][0]){
                  printf("la maquina ha ganado");
              }
          i=i-1;
              //turno de las negras
          }

      }
```

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

```
//CODIGO B
#include <stdio.h>
#include <iostream>
using namespace std;
int main()
{
    int Peon[7][7];
    int i=3;
    while(not(Peon[i][0]==Peon[0][0])  or  not(Peon[i-1][0] != NULL)){
              Peon[i][0] = Peon[i-1][0];
              if(Peon[i-1][0] == Peon[0][0]){
                  printf("la maquina ha ganado");
              }
              i=i-1;
              //turno de las negras
          }
      }
```

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

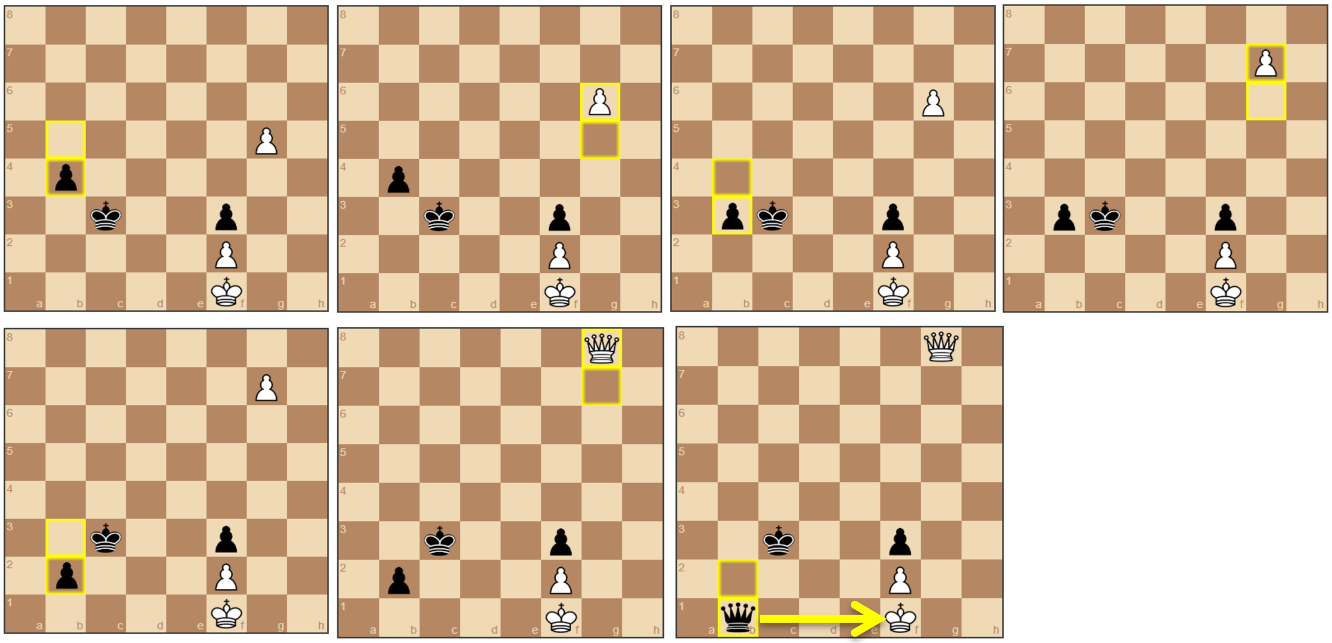
ii. ¿qué es lo que están haciendo los códigos A y B? explica su salida

iii. ¿Crees que es conveniente para la maquina el haber preferido utilizar alguno de los dos códigos? ¿Por qué?

iv. Para el código B, ¿si quitamos la condición; $\neg(\boldsymbol{Peon}[\boldsymbol{i}-\mathbf{1}][\mathbf{0}] \,!= \,\boldsymbol{NULL}$, dentro del ciclo $\boldsymbol{while}$, consideras que se pueda seguir ejecutando el juego para ese caso particular? ¿Por qué?

b) segundo caso.

En este otro diagrama se muestra una partida de ajedrez, donde en este caso juegan dos máquinas y el turno es para las negras, ambos bandos solo pueden realizar el movimiento más conveniente el cual es tratar de coronar al peón salvo que alguno de los dos cometa algún error.



A continuación se muestra la implementación de los códigos **C** y **D** en el lenguaje C++ de la partida anterior, usa el diagrama para ayudarte y ve marcando sobre este las iteraciones que creas conveniente**.**

```
//CODIGO C
#include <stdio.h>
#include <iostream>
using namespace std;
int main()
{
  int peonNegro[7][7];
  int i;
  for(i=3;not(i<3 or i>6);i++){
          peonNegro[i][1] = peonNegro[i+1][1];

          if(peonNegro[i][1] == peonNegro[7][1])
              printf("jaque mate");
        }

      }
```

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

```
//CODIGO D
#include <stdio.h>
#include <iostream>
using namespace std;
int main()
{
  int peonNegro[7][7];
  int i;
  for(i=3;(i>=3 and i<=6);i++){
          peonNegro[i][1] = peonNegro[i+1][1];

          if(peonNegro[i][1] == peonNegro[7][1])
              printf("jaque mate");
      }
      }
```

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

i. ¿Cuál es el resultado que obtuviste con los códigos C y D?

ii. ¿Crees que es conveniente para la maquina el haber preferido utilizar alguno de los dos códigos? ¿Por qué?

iii. ¿Encuentras alguna relación entre los códigos A y B que analizaste anteriormente que con los códigos C y D?

Sección 3.

i. Completa las siguientes tablas.

Tabla A

A	B	A and B	not (A and B)	not A and not B
Verdad	Verdad	Verdad		
Verdad	Falso	Falso		
Falso	Verdad	Falso		
Falso	Falso	Falso		

Tabla B

A	B	A or B	not A or not B	not (A or B)
Verdad	Verdad	Verdad		
Verdad	Falso	Verdad		
Falso	Verdad	Verdad		
Falso	Falso	Falso		

ii. ¿Qué relación pudiste encontrar entre las últimas dos columnas de la tabla A y la tabla B, encuentras alguna similitud?

iii. **Relaciona la salida esperada de cada código con su respectiva tabla de verdad y anótalos en la siguiente tabla.**

TABLA	CODIGO

Tabla 3

A	B	(A∧B)
1	1	1
1	0	0
0	1	0
0	0	0

Tabla 2

A	B	¬A∧¬B
1	1	0
1	0	0
0	1	0
0	0	1

Tabla 1

A	B	(A∨B)
1	1	1
1	0	1
0	1	1
0	0	0

Tabla 4

A	B	¬(A∧B)
1	1	0
1	0	1
0	1	1
0	0	1

```
#include <stdio.h>
int main(){
int a[]={1,1,0,0};
int b[]={1,0,1,0};int i,s[4];

     //CODIGO 3
      for(i=0;i<4;i++){
        if (a[i]==1 or b[i]==1)
          s[i]=1;
              else
                s[i]=0;
```

```
            }
        for(i=0;i<4;i++)
                printf("\n\t%d\t%d\t%d",a[i],b[i],s[i]);

        //CODIGO 1
        for(i=0;i<4;i++){
        if(a[i]==1 and b[i]==1)
                    s[i]=1;
                else
                    s[i]=0;
            }
            for(i=0;i<4;i++)
                printf("\n\t%d\t%d\t%d",a[i],b[i],s[i]);

        //CODIGO 4
        for(i=0;i<4;i++){
        if(a[i]==1 and b[i]==1)
                    s[i]=1;
                else
                    s[i]=0;
            }
        for(i=0;i<=4;i++){
             if(a[i]==1 or b[i]==1)
                 s[i]=0;
             else
                 s[i]=1;
        }
        for(i=0;i<4;i++)
             printf("\n\t%d\t%d\t%d",a[i],b[i],s[i]);


        //CODIGO 2
        for(i=0;i<4;i++){
             if(a[i]==1 or b[i]==1)
                s[i]=1;
             else
                s[i]=0;
            }
        for(i=0;i<=4;i++){
             if(a[i]==1 and b[i]==1)
                 s[i]=0;
             else
                 s[i]=1;
        }
        for(i=0;i<4;i++)
             printf("\n\t%d\t%d\t%d",a[i],b[i],s[i]);
        }
```

Actividad 2.

Introducción:

En la actividad anterior se estuvo trabajando con un par de propiedades que son empleadas muy recurrente en lógica y programación, las cuales tiene por nombre las leyes de De Morgan.

Sección 1.

i. Completa la tabla desarrollando para cada caso las leyes de De Morgan respectivamente.

$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{not(A\ \ and\ B)}$	$\boldsymbol{not(A\ or\ B)}$
$x < a$	$x >= b$		
$x > b$	$x == b$		
"Juan es cocinero"	***"pedro es policía"***		
$\neg(R)$	$\neg\ (S)$		
(R)	$\neg\ \neg\ (S)$		
$a\ != b$	$a\ != c$		

Sección 2.

A continuación, Analiza el siguiente código.

```
1. //CODIGO E
2. #include <stdio.h>
3. #include <iostream>
4. using namespace std;
5. int main()
6. {
7.       bool A[] = {true, false};
8.       bool B[] = {true, false};
9.       bool C[4] = {};
10.      bool D[4] = {};
11.      bool iguales = true;
12.      int contador=0;
13.
14.          for(int x=0;x<2;x++){
15.              for(int y=0;y<2;y++){
16.                C[contador] = not(A[x] or B[y]);
17.                D[contador] = not(A[x]) and not(B[y]);
18.                cout<<C[contador]<<"|"<<D[contador]<<endl;
19.                   contador = contador + 1;
20.               }
21.          }
22.          for(int x=0;x<contador;x++){
23.                if(C[x] != D[x])
24.                      iguales = false;
25.                      break;
26.          }
27.          if(iguales)
28.                      cout << "se cumple la ley de Morgan" <<endl;
29.          else
30.                      cout << "No se cumple la ley de Morgan" <<endl;
31.      }
```

i.

1. **¿cuál será la salida del código E?**

2. **¿está presente alguna ley de De Morgan en el código E? señala en que líneas son las que evalúan esta ley.**

3. **¿alguna de las leyes no está presente? ¿cuál?**

ii. Si respondiste que falta alguna ley en el programa anterior, ¿qué líneas cambiarias en el código para que se pueda ejecutar la otra ley de De Morgan?

Escribe tu versión de esas líneas,(puedes escribir solamente las líneas que consideres):

iii. completa la expresión de la línea 15 del código F para que el código pueda ejecutarse y se cumpla la expresión.

```
//CODIGO F
#include <stdio.h>
#include <iostream>
using namespace std;
int main(){
        bool R[] = {true, false};
        bool H[] = {true, false};
        bool C[4] = {};
        bool D[4] = {};
        bool iguales = true;
        int contador=0;

      for(int x=0;x<2;x++){
       for(int y=0;y<2;y++){
          C[contador] = not(not(R[x]) and not(H[y]));
          D[contador] = R[x] and H[y];
          contador = contador + 1;
       }
      }
          for(int x=0;x<contador;x++){
            if(C[x] != D[x])
                iguales = false;
                break;
          }
          if(iguales)
              cout << "se cumple la expresión" <<endl;
          else
              cout << "No se cumple la expresión" <<endl;

      }
```

Sección 3.

i. Aplica las leyes de De Morgan a los siguientes ejercicios.

1. $not(q_1\ and\ \ q_2) \equiv$

2. $not(p_1\ and\ \ p_2\ \ and\ \ p_3) \equiv$

3. $not(not\ (r_1)\ and\ \ not(r_2)\ \ and\ \ not(r_3)) \equiv$

4. $not(s_1\ or\ \ s_2\ \ or\ \ s_3\ \ or\ \ s_4\ ) \equiv$

5. $not(not(t_1)\ \ or\ \ not(t_2)\ \ or\ \ not(t_3)\ \ or\ \ not(t_4)\ ) \equiv$

ii. Según tus respuestas de la parte anterior:

1. ¿Observas algún patrón en los ejercicios anteriores?

2. Compara las expresiones $not(p_1\ and\ \ (p_2\ or\ \ p_3))$ y $not((p_1\ and\ \ p_2)\ or\ \ p_3))$.

¿las expresiones anteriores son equivalentes?

iii.

1. Propón expresiones para la negación de la disyunción y la conjunción para el caso de n premisas de acuerdo a las leyes de De Morgan.

2. Explica verbalmente que significan esas expresiones.