

Windows 상에서의 Flowchart – C 코드 변환 시스템

Flowchart – C code Conversion System on Windows

주윤희, 이창우, 손영선
동명정보대학교 정보통신공학과

Yun-Hee Ju, Chang-woo Lee and Young-Sun Sohn
Dept. of Information & Communications Engineering
TONGMYONG University of Information Technology
E-mail : yssohn@tmic.tit.ac.kr

ABSTRACT

본 논문에서는 사용자가 Windows 상에서 자신의 알고리즘을 반영시킨 flowchart를 작성하면 그에 해당하는 C프로그램을 생성시켜 주는 시스템을 구현하였다. 사용자가 그린 flowchart 기호들과 흐름선 및 기호 안에 기술된 문자정보를 연결형 리스트를 사용하여 표현하였다. 기술된 문자 정보를 검색하여, 사용된 함수에 따라 해당 헤더 파일을 추가하고, 입·출력 함수에 대한 변환을 행한다. 그려진 flowchart기호에 대하여 흐름선의 개수나 연결기호를 이용한 변환 알고리즘을 적용하여 C코드로 변환하여 화면상에 출력시키고, 파일로 저장하였다. 본 시스템을 다양한 알고리즘에 적용시켜서 만족한 결과를 얻을 수 있었다.

Key Words : C언어, flowchart, 연결형 리스트, C 코드 변환 알고리즘

1. 서론

소프트웨어 개발 환경에서 소프트웨어 품질 향상과 생산성 증대에 가장 큰 영향을 주는 것은 개발 기법과 개발 도구이며, 최근에는 개발 도구에 관심이 점점 커지고 있다[1~3]. 전문적인 프로그래머 뿐만 아니라 비전문가도 쉽게 프로그래밍 할 수 있는 뛰어난 자동화 도구들의 개발이 필수적인 것으로 받아들여지고 있다. 자동화 도구는 사용자의 알고리즘만으로 사용자의 소프트웨어 개발 행위를 대신해 주어서 보다 편리한 소프트웨어 개발 환경을 지원해주는 것이다. 그러나 이러한 자동화 도구들은 사용자가 시스템의 사전지식 및 절차를 숙지해야 하는 어려움이 있으며, 골격 프로그램 생성에 그치고 마는 경우가 대부분이다[4~9].

본 논문에서는 이러한 문제들을 해결하기 위하여 조작성이 간단하며, 사용자가 Windows 상

에서 자신의 알고리즘을 반영시킨 flowchart를 작성하면 그에 해당하는 C 프로그램을 생성시켜 주는 시스템을 구현하였다.

2. 시스템 알고리즘

본 논문의 전체 시스템 알고리즘은 그림 1의 시스템 흐름도에서 알 수 있듯이 사용자가 Windows 상에서 자신의 알고리즘이 나타나도록 flowchart 기호와 흐름선, 기호 속 문자열 등을 작성하면, flowchart를 구성하는 정보들이 그림을 저장하는 연결형 리스트로 저장된다. 저장된 정보에 대하여 검토와 수정을 거친 뒤 C 변환을 위한 연결형 리스트의 각 노드에 필요한 정보만을 전달한다. C 변환 연결형 리스트에서는 각각의 기호와 기호 속 문자열 정보, 흐름선 등을 이용한 C 코드 변환 알고리즘을 바탕으로 C 변환 하게 된다. 모든 정보에 대한

변환이 끝나면 출력창으로 C 코드를 나타내어 주며 파일로 저장하게 된다. 저장된 C 프로그램 파일을 선택하여 컴파일 및 실행할 수 있다.

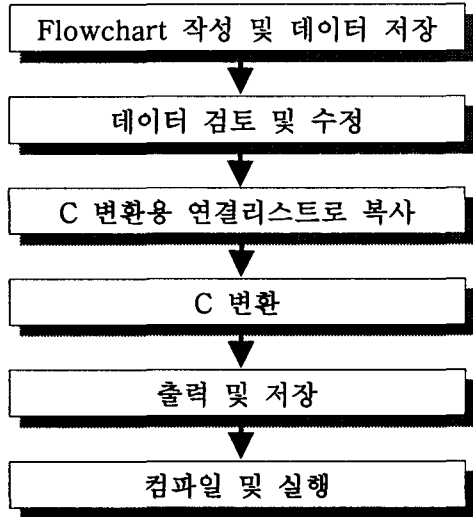


그림 1. 전체 시스템 흐름도

3. Flowchart 작성 및 처리

사용자가 그림 2와 같이 자신의 알고리즘에 맞게 flowchart를 작성한다. 알고리즘이 표현된 flowchart를 작성하는데 있어서 흐름선과 기호가 이어지는 부분이 떨어져 있거나 흐름선이 완전하게 수직, 수평이 아니더라도 시스템이 흐름을 인식하도록 처리하였다. flowchart의 구성 요소를 하나씩 작성하면 그에 대한 정보가 flowchart 그림을 저장하는 연결형 리스트의 각 멤버 변수에 저장되도록 하였다.

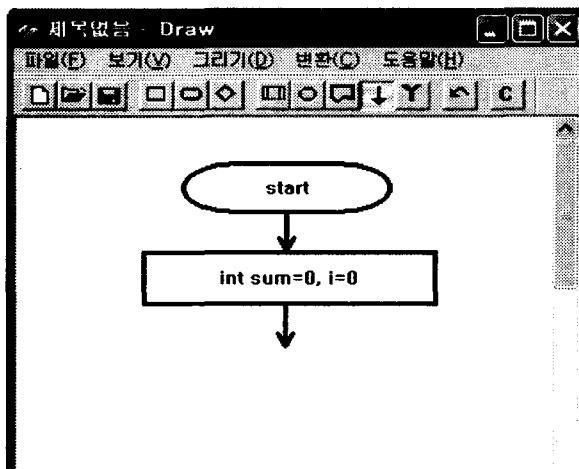


그림 2. flowchart 작성 모습

연결형 리스트에 저장된 flowchart의 정보를 C 변환용 연결형 리스트로 전달하기 위해서 일부 정보에 대한 가공이 필요하다. 흐름선이나 연결기호가 어떤 flowchart 기호와 이어져 있는지, 흐름선의 참, 거짓을 나타내주는 Y(Yes), N(No) 문자가 어느 흐름선에 속하는지를 정해주는 부분이다.

흐름선의 경우 시작과 끝 부분의 좌표점이 각 flowchart 기호의 상, 하, 좌, 우 중 어느 부분과 이어져 있는지를 검사하여 판단한다. 연결기호 역시 마찬가지로 모든 flowchart 기호를 작성한 차례대로 검사하여 어느 기호와 흐름선으로 이어져 있는지를 판단한다. Y나 N 문자가 어느 흐름선에 속하는지를 설정한 방법은 문자와 모든 흐름선의 중점과의 거리를 비교하여 가장 가까운 거리에 있는 흐름선에 Y나 N 문자를 흐름선과 묶어주도록 하였다.

연결형 리스트에 저장된 정보의 검토 및 수정이 끝나면 flowchart의 구성 요소를 각각의 C 변환용 연결형 리스트의 노드로 전달하는데, 각 노드는 하나의 기호와 그에 연결된 흐름선, 연결기호 정보, 기호 속 문자열 정보로 이루어져 있다. 노드에 맞게 수정된 정보를 전달하면 작성된 flowchart의 기호 중 연결기호를 제외한 터미널, 처리, 선택, 함수, 파일 기호의 개수 만큼의 노드가 만들어지게 된다.

4. C 변환

C 변환 연결형 리스트의 정보에 대한 C 변환 과정은 함수에 대한 헤더 파일 추가, 변수 선언 체크, 입출력 함수 및 flowchart 기호에 대한 변환, 들여쓰기 설정의 순으로 이루어진다.

4.1 각종 함수에 대한 헤더 파일 추가

그림 3과 같이, 각 노드에 저장된 문자열 데이터를 모두 검색하여 '('가 발견 되면 '('앞의 문자열을 추출한 다음, 대표적 헤더 파일 중에서 추출된 문자열이 있는지 검색 한다. 추출된 문자열이 발견되는 해당 헤더 파일 명을 추가 시킨다.

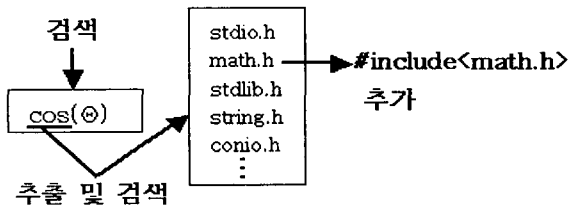


그림 3. 헤더 파일 추가

4.2 변수 선언 체크

자료형이 선언 되지 않은 변수에 대해서는 기본적으로 int형으로 선언하고 0으로 초기화 시킨다.

4.3 입·출력 함수에 대한 변환

그림 4의 ①과 같이, 각 노드에 저장된 문자열 데이터 중 'print'나 'input'이 있는지 검사한다. 'print'나 'input'이 발견되면 출력 또는 입력할 자료형을 알기 위해 ②와같이 헤더 노드 쪽으로 이동하면서 선언부를 찾는다. 선언된 자료형에 맞게 입·출력 문을 ③과 같이 변환하여 저장한다.

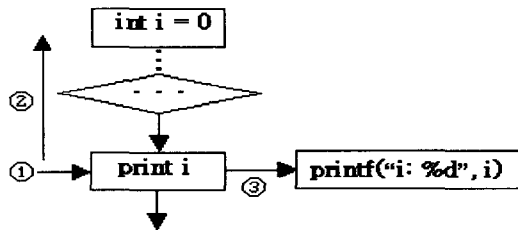


그림 4. 입·출력 함수 변환

4.4 Flowchart 기호에 따른 변환

각각의 flowchart기호의 종류와 기호명은 표 1과 같다.

표1. flowchart 기호

flowchart 기호	기호명
	터미널
	처리
	선택
	함수
	파일
①	연결

4.4.1 터미널 기호

flowchart의 시작과 끝부분을 나타내는 start, stop으로 구분되어 진다. 시작을 나타내는 start를 int main()으로 변환 시켜 주고 끝을 나타내는 stop을 return 0으로 변환시켜준다.

4.4.2 처리 기호

보통의 경우는 문자열 다음에 세미콜론만 추가 시켜 주나 처리기호 바로 앞 노드의 정보가 흐름선이 2개인 선택기호이고 아래 흐름선이 N인 경우는 else문으로 변환시켜 준다. 왼쪽 또는 오른쪽에서 입력되는 흐름선이 존재하는 경우는 do~while문으로 변환 시켜 준다.

4.4.3 선택기호

연결된 흐름선의 개수를 파악하여, 흐름선의 개수가 2개인 경우에는 if문이나 else if문으로 변환 시켜 주고 흐름선의 개수가 3개인 경우에는 while문으로 변환시켜 준다.

4.4.4 함수기호

예를 들어, 그림 5의 ①과 같이 함수기호가 저장된 연결리스트의 노드부터 저장된 문자열을 모두 검사하여 'return'을 찾는다. 만약 'return'을 발견하지 못하면 void 함수명()을 각각 함수선언과 함수정의 부분에 변환 시켜 주고 함수선언에는 세미콜론을 추가해 준다. ②와 같이 'return'을 발견하면 'return' 다음의 문자열을 추출하여 ③과 같이 문자열의 자료형이 무엇인지를 찾은 다음 int imax(int n, int m)와 같이 변환 시켜 주고 함수 선언에 세미콜론을 추가시켜 준다.

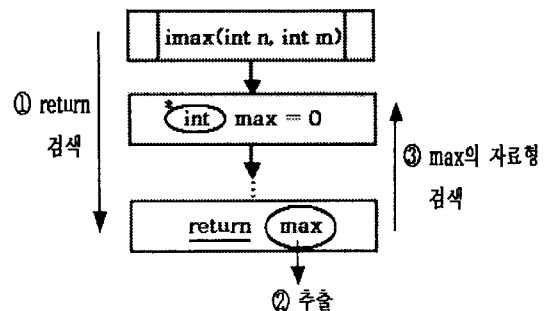


그림 5. 함수 처리

4.4.5 파일 기호

기호 속에 명시된 파일을 열어 파일 안의 내용을 화면에 출력 해주는 코드를 추가해준다.

4.4.6 연결기호

위나 아래의 연결기호 번호와 비교하여 특정 구문의 끝이나 시작으로 판단하여 {, }로 변화시켜 준다.

4.5 들여쓰기 설정

C 변환용 연결리스트의 모든 노드에 대한 변환이 끝나면 변환된 코드에서 {, }를 검색하여 줄바꿈 후의 삽입할 '\t' 개수를 증가 또는 감소 시켜 들여쓰기를 설정해 준다.

5. 출력, 저장 및 컴파일

C 변환 연결형 리스트의 모든 노드에 대한 변환이 끝나면 그림 6과 같이 변환된 C 코드를 Dialog 창으로 출력시킨다. 출력된 C 코드를 편집하거나 C 프로그램 파일로 저장할 수 있도록 시스템을 구성하였다. 저장된 C 파일을 Microsoft Visual C++ 컴파일러를 이용하여 컴파일하고 실행까지 가능하도록 하였고, 또한 이미 작성한 C 파일을 로드하여 컴파일 및 실행을 할 수 있는 시스템을 구현하였다.

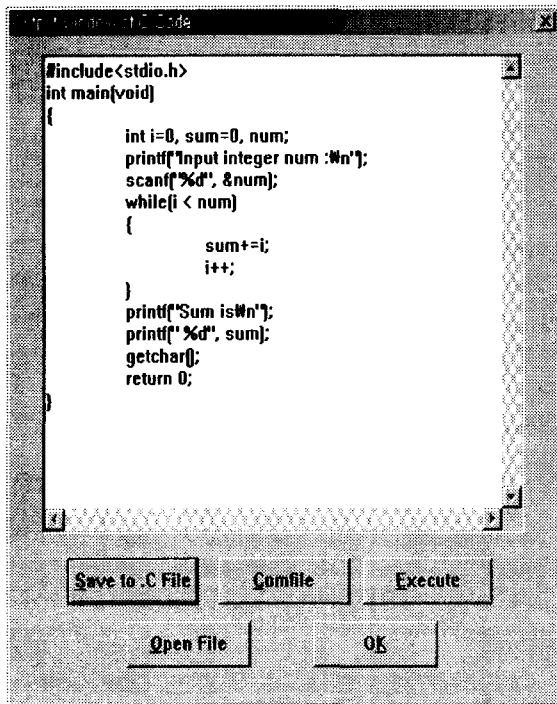


그림 6. C 코드 출력화면

6. 결론 및 향후과제

본 논문에서는 사용자가 Windows 상에서 자신의 알고리즘을 반영시킨 flowchart를 작성하면, 그에 해당하는 C프로그램을 생성시켜 주는 시스템을 구현하였다. 연결형 리스트를 사용하여 작성한 flowchart에 대한 정보를 저장하였으며, C변환 알고리즘을 설계하여 그에 맞게 flowchart를 C 코드로 변환시켜 출력 및 파일로 저장하였다.

향후과제로는 구조체나 포인터 등을 이용하는 복잡하고 다양한 알고리즘이 표현된 flowchart에 대한 C 코드 변환 시스템의 구현이 고려되어진다.

참고문헌

- [1]권호열, "소프트웨어 개발 프로세스의 연구 동향", 한국 정보 과학회지, Vol. 20, No. 1, PP. 0006~0014, 2000
- [2]정목동, "자동 프로그래밍 시스템의 구현과 효율적인 프로그램 명세 언어", 外大論叢, Vol. 6, No. 1, 1988
- [3]하수철, "소프트웨어 개발 환경에서의 시각 프로그래밍", 대전대학교 산업기술연구소 논문집, Vol. 4, No. 5, PP. 69~82, 1993
- [4]최승교, 주영상, 이진영, "소프트웨어 개발 자동화 도구를 위한 다이어그램 형식과 구현", 三陟大學校 論文集, Vol. 31, No. 1, PP. 207~231, 1998
- [5]정선환 "COBOL 프로그래밍을 위한 자동화 도구의 설계 및 구현", 建國大學校 産業大學院, Vol. 5, No. 1, PP. 67~100
- [6]최근수, 하상호 "C 해석기를 통한 C 프로그램 실행의 시각화", 정보처리학회, Vol. 9, No. 2, PP. 297~300, 2002
- [7]곽상훈, 이동익, 배영환 "실시간 운영체제 환경을 고려한 SDL명세로부터의 C코드 생성", 정보과학회, Vol. 28, No. 02, PP. 469~471, 2001
- [8]김병건, 최원혁, 김성재, 김승호, "SDL에서 C로의 번역기의 설계 및 구현", 정보과학회, Vol. 28, No. 01 PP.0574~0576, 2001
- [9]선은두, 김희천, 우치수, "구조적 설계를 통한 C코드 생성", 정보과학회, 1994