2004년도 대한전기학회 하계학술대회 논문집 2004.7.14-16

Lex와 Yacc를 이용한 복구 절차 생성을 위한 그룹핑 알고리즘 개발에 관한 연구

김인수*, 김경근, 최영준, 홍정기
(주) 효성 중공업연구소

A Study on the Development of Grouping Algorithm for Making Sequence of Power System Restoration using Lex and Yacc

In-Su Kim*, Kyung-Geun Kim, Young-Jun Choi, Jung-Ki Hong
Hyosung Corporation, Power System Automation Team, Power & Industrial Systems R & D Center

Abstract - 전력 계통은 거대한 네트워크화되고 있으며, 전력계통을 신뢰성 있게 운용하는 것이 어려워지고 있다. 이에 따라 전력계통에 대한 숙련된 운전기술이 필요하게 되고, 실제 전력계통 모의를 통한 미경험 고장의 처리기술을 습득해야할 필요성이 생기게 되었다. 그러한 취지에서 각종 유틸리티는 운용자 지원 소프트웨어나 고장의 발생과 복구를 훈련할 수 있는 시뮬레이터를 대부분 구비하고 있다. 이러한 프로그램의 핵심적인 기능은 고장의 시나리오를 실제 계통에서 발생할 수 있는 고장으로 상정하고, 전문가 그룹에 의해 도출된 복구절차에 의해 복구 과정을 훈련하는 것이다. 전력계통이 복잡화되고 거대화됨에 따라 복구절차 또한 복잡화되고 있으므로 기존의 Case-by-case에 의해 복구절차를 생성하는 데는 한계가 있다. 그러므로 본 연구에서는 Lex라는 구문분석기와 Yacc라는 파서를 이용하여 복구절차 생성시 복구 절차를 적절한 그룹을 만들고, 그 그룹을 통해서 조합 가능한 모든 복구 조작 순서를 생성하기 위한 그룹핑 알고리즘을 구현하고자한다.

1. 서 론

지난 2003년 8월 14일, 미국 동부 및 케나다 일부지역의 전력 과부하로 인한 정전사고는 오늘날의 전력계통이 얼마나 복잡하게 구성되어 있는지와 사고의 파급효과가 얼마나 커다란지를 보여주는 단적인 예일 것이다. 전력계통에서의 고장은 인위적이든 비인위적이든 피할 수 없는 것이고, 관건은 사고를 신속하게 검출하고, 정확하게 판별하여 이에 합당한 대처를 해나가는 것일 것이다. 즉, 파급효과를 최소화하는 방향으로 고장 계통을 복구하는 것이 필요한 것이다. 이를 위해서는 해당 분야의 전문가의 의견에 따라 고장을 복구해야 할 것이다. 한편, 전력계통에서의 사고는 빈번한 일이 아니므로 전력계통 운영자는 미경험 고장에 대처해야 할 경우가 있을 것이다. 이에 따라 신속하고도 신뢰성 있는 복구를 위한 운용자 지원시스템이나 고장 발생 및 복구 훈련 프로그램에 대한 요구가 심화되고 있는 실정이다.

따라서 본 연구에서는, 전문가 그룹에 의해 도출된 복구 절차를 구성함에 있어서 복구절차를 적절할 그룹으로 생성하여 가능한 모든 복구절차의 조합을 자동으로 생성하는 데 그 목적으로 둔다. 이를 위해서, Lex라는 구문분석기와 Yacc라는 파서를 사용하였다.

2. lex와 yacc

2.1 정규 표현식

표현식(expression)이란 문자 그대로의 의미 이상으로 해석될 수 있는 메타문자(meta-characters)라고 부르는 문자들의 집합이라 할 수 있다. 예를 들어, 인용 부호(quote mark)는 어떤 사람이 말한 것을 나타내 주기도 하지만 또한 그 뒤에 나오는 심볼에 대해서 메타적 의미를 부여하기도 한다. 어떤 메타문자도 포함하지 않아서 문자열 자체의 뜻을 그대로 갖고 있는 문자열이 가장 간단한 형태의 정규 표현식이고 주로 텍스트 탐색과 문자열 조작에 쓰인다. 이러한 정규 표현식은 하나의 문자와 일치하거나, 혹은 문자열의 일부분(substring)이나 전체 문자열인 문자 집합들과 일치하게 된다. 본 연구에서는 일반적인 정규 표현식에 따르는 문자열을 입력으로 받아서 처리하게 된다[1].

2.2 lex와 yacc

일정한 룰에 따라 입력된 내용을 변환하는 프로그램을 만드는데 유용하게 사용되는 툴이 lex와 yacc이다. 일정한 룰에 따르는 입력을 받아들이는 프로그램에서 반복하는 작업은 입력에서 의미 있는 단위로 분해하여 그들 사이의 관련성을 얻어내는 것이다. 이렇게 임의의 입력을 토큰(token)이라고 하는 의미있는 단위로 나누는 작업을 대개 어휘 분석(lexical analysis) 혹은 렉싱(lexing)이라고 한다. lex는 토큰들을 받아들여 주어진 토큰을 식별할 수 있는 C 루틴을 생성한다. 이 루틴을 어휘 분석기(lexical analyzer), 렉서(lexer) 혹은 스캐너(scanner)라고도 한다. 한편, lex에 기술하는 내용을 렉스 명세(lex specification)라 한다. lex에 사용되는 토큰 설명(token description)을 전술한 정규 표현식이라 하며 lex는 찾고자 하는 표현식의 개수에 관계없이 lexer가 입력 문자열을 빠르게 검색할 수 있는 형태로 이러한 정규 표현식을 바꾸어 준다. 파싱(parsing)은 입력된 내용이 토큰으로 나누어진 후 토큰 사이의 관련성을 확립하는 것이며, 문법(grammar)은 그 과정에서 프로그램이 이해할 수 있는 관련성을 정의하는 규칙이다. yacc는 문법에 대한 설명을 받아들여 그 문법을 파싱할 수 있는 루틴을 생성해주는데, 이 루틴을 파서(parser)라 한다. 일반적으로, yacc에 의해 생성된 파서는 C 문법에 정의된 규칙과 일치하는 일련의 입력 토큰을 인식하고 일치하지 않는 토큰에 대해서는 구문 에러(syntax error)를 발생시킨다. 대개 yacc가 생성한 파서는 직접 작성하는 파서보다 빠르지 않다는 단점을 가지고 있지만 파서를 쉽게 작성하고 수정할 수 있다는 장점이 있다. 복구절차 생성에 있어서 소비되는 시간은 중요한 문제가 아니므로 본 연구에서는 lex와 yacc를 사용하여 알고리즘을 구현하였다[2].

3. 복구순서의 탐색

전력계통의 사고를 복구하는 과정은 사고구간, 정전구간, 그리고 전력계통의 실시간 정보를 받아들여 복구 순서를 결정하는 것이라 할 수 있다. 전력계통의 사고는 정전시간 및 정전구간이 최소화하여 파급효과를 최대한 줄이는 것이 관건이다. 즉, 전력계통의 사고복구는 최적화된 해를 찾는 문제이다. 그런데 전력계통에 사고가 났을 때 N의 복구 과정이 있을 때, 생각할 수 있는 복구 방법은 N!개 이며 이를 모두 비교/판단하여 최적을 결정한다면 N이 커질 때 탐색에 소요되는 비용은 폭발적으

로 증가한다. 그러므로 해를 구하고자 하는 탐색에 있어서는 탐색을 제한하는 것이 매우 중요하다. 일반적으로 문제를 표현하는 방법은 AND/OR 트리와 상태 공간 모형을 사용할 수 있다. 어떤 문제를 상태공간으로 형성하기 위해서는, 우선 문제의 상태가 무엇인지를 살펴보아야한다. 전력계통 사고복구 문제는 하나의 복구절차가 진행된 형태를 문제의 상태로 이해 할 수 있다. 문제의 상태를 나타내기위해서는 원칙적으로, 어떤 형태의 데이터 구조로도 가능하다. 예를 들어, 심볼 스트링 벡터, 2차원 배열, 트리, 리스트 등을 생각할 수 있다. 단지 문제에 따라 상태표현에 자연스럽고, 운영에 효율적인 것을 고르는 것이 중요하다. 복구순서를 결정하는 문제는 다음의 그림1과 같이 표시될 수 있다.

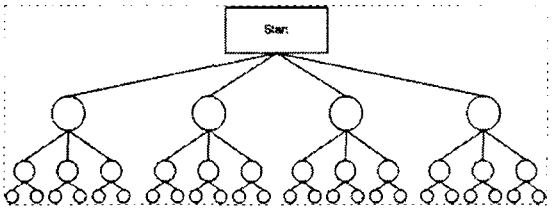

그림 1. 전력계통 사고복구 문제의 상태공간

위 그림에서 원은 하나의 선택을 나타낸다. 즉 가장 위의 네 개의 원은 진행되어야할 절차의 수가 4개임을 가리키고 있다. 이때 복구의 순서를 정하는 작업은 각 단계의 원을 거쳐 마지막 단의 원까지 도달하면 끝난다. 즉, 한가지의 복구순서가 결정된 것이다. 이것은 복구되어야할 절차의 나열을 의미한다. 위의 그림에서 최적의 경로를 찾는다면 모두 4+4*3+4*3*2개의 경로를 탐색하여야 한다. 이때 모두 N개의 복구절차가 존재하면 N!개의 목표상태가 생기며 그 경로의 수는 다음과 같다[3].

$$cnt(path) = \sum_{K=1}^{N} \frac{N!}{K!}$$

그러나, 이를 모두 탐색하는 것은 매우 어렵고 비용 또한 상당히 크다. 그러므로 탐색을 제한하여 탐색해야할 경로의 수를 줄이는 것이 필요하다. 본 연구에서는 다음과 같은 탐색 전략을 세우고 목표상태가 얻어질 때까지 연산을 수행하도록 하였다. 전문가 그룹에 의해서 도출된 복구 절차를 입력으로 받아들이면서 탐색을 제한하기 위해서 논리 연산에서의 AND와 OR 개념을 도입하여 AND로 연결된 복구절차는 순서를 변경할 수 없고 OR로 연결된 복구절차는 순서에 무관한 것으로 정의하였다. 이를 통해서 탬색을 제한하며 가능한 모든 복구절차 순서의 나열을 우선 목표로 선정하였다.

4. 그룹화 알고리즘

본 연구에서는 앞서 얘기한 바와 같이 Lex와 Yacc라는 구문 분석기와 파서를 사용하여 그룹화 알고리즘을 구현하였다. 전문가 그룹에 의해서 여러 복구 절차가 제안되었을 때 타당한 복구 절차의 조합을 만들기 위해서는 우선 이러한 복구 절차들을 정규 표현식을 따르는 문장으로 변환해야한다. 변환된 문장을 Lex에 의해 만들어진 구문 분석기를 통해 의미있는 단위인 토큰으로 뽑아오게 되고 그 토큰들 사이에 적절한 관계성을 찾기 위해 Yacc에 의해 만들어진 파서를 수행시키게 된다. 최종적으로 생성된 가능한 복구 절차의 순서가 화면에 표시되면서 알고리즘의 사용이 종료된다. 다음 그림 2는 이러한 알고리즘의 순서를 나타내고 있다.

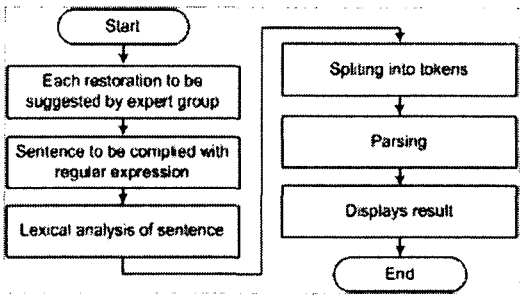

그림 2. 그룹화 알고리즘의 순서도

5. 그룹화 알고리즘의 검토

그룹화 알고리즘을 검토하기 위해, 사고가 발생했을 때 다음의 R1, R2 및 R3의 조작을 전문가 그룹이 제안했을 때, 다음 표 1에서처럼 제안했을 때를 고려하였다. 다음 그림 3은 그룹화 알고리즘을 통해 얻은 각각의 Case 별로 가능한 복구 조작의 순서를 나열한 것이다.

표 1. 전문가 그룹에 의해 제안된 복구조작-Case ①

Case	제안
1	R1, R2, R3를 순서대로 조작 가능
2	R1, R2, R3를 순서없이 조작 가능
3	R1-R2를 순서대로 조작, R3 무관
4	R2-R3를 순서대로 조작, R1 무관
5	R3-R1를 순서대로 조작, R2 무관

이러한 Case들은 AND/OR을 통해 탐색을 제한하는 기본적인 사례들이라 할 수 있다.

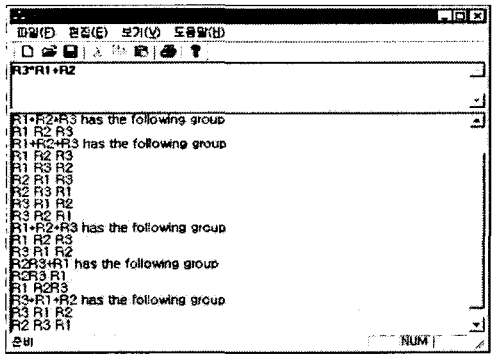

그림 3. 제안된 알고리즘의 결과 출력

그러나 실제 계통에서 이전의 사례같이 단순한 경우는 찾아보기 힘들기 때문에 좀더 실제 계통에서 발생할 수 있는 사례를 통해 검증해보았다. 표 2처럼, 전문가 그룹에 의해 제안된 복구절차가 CB1-CB2의 순으로 개방해야하는 경우와 CB2-CB1의 순으로 개방해야하는 경우 및 CB1과 CB2를 순서에 관계없이 개방하는 것일 때를 고려하였다.

표 2. 전문가 그룹에 의해 제안된 복구조작-Case ②

Case	제안
1	CB1, CB2를 순서대로 개방
2	CB2, CB1을 순서대로 개방
3	CB1, CB2를 순서없이 개방

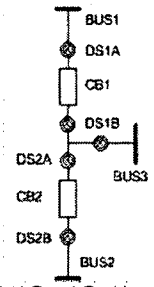

그림 4. 1.5 CB 방식을 사용하는 계통에서의 CB 조작

CB를 개방하고 연결된 두개의 DS를 개방하는 것은 일반적으로 순서에 상관없기 때문에 본 Case에서는 고려하지 않았고 그림 5는 제안된 알고리즘을 통해 얻어진 결과를 나타낸다.

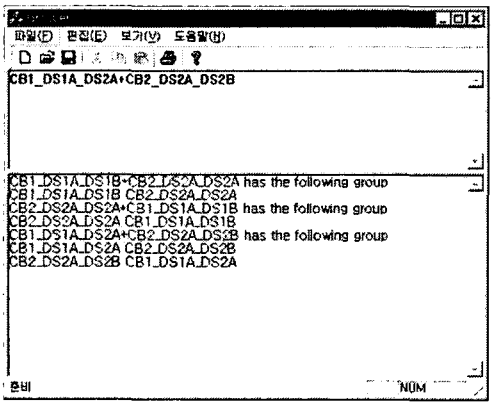

그림 5. 제안된 알고리즘의 출력 결과

6. 결 론

전력 계통이 점차 복잡화되고 거대화됨에 따라 전력계통에서의 사고는 커다란 파급효과를 가진다. 관건은 얼마나 사고를 신속하게 검출하고 합당한 대처를 통해서 파급효과를 최소화하느냐 이다. 이를 위해서는 해당 분야의 전문가 그룹에 의해 도출된 복구 절차를 데이터베이스화하여 전력계통 운영자에게 미경험 고장에 대처할 수 있도록 운용자 지원 시스템이나 고장 발생 및 복구 훈련 프로그램을 구비해야 할 것이다. 본 연구에서는 이처럼 전문가 그룹에 의해서 제안된 복구 절차를 입력받아 가능한 모든 조합의 복구 절차를 생성시켜주기 위한 그룹화 알고리즘을 개발하였다. 하지만, 그림 5에서 알 수 있듯이 제안된 알고리즘을 통해 얻어진 결과를 화면에 출력해주는 기능은 있으나 이를 데이터베이스화 할 수 있는 기능은 없다. 향후, 좀더 향상된 전문가 시스템이나 운용자 지원 시스템에 본 알고리즘을 적용하기 위해서는 전문가 시스템에서 필요로 하는 데이터베이스나 지식베이스와의 인터페이스 문제를 해결해야 할 것이며, 실제로 조작되고 있는 복구절차를 통해 제안된 알고리즘을 적용해볼 필요가 있을 것이다.

[참 고 문 헌]

[1] John R. Levine, Tony Mason, Doug Brow, "Lex & Yacc", O'Reilly & Associates, October, 1992.
[2] JEFFREY E. F. FRIEDL, "Mastering Regular Expressions", O'Reilly & Associates, March, 1997.
[3] "전력계통 사고관정 및 복구지원 전문가 시스템에 관한 연구", 과학기술처, pp.61 62, 1990. 8.
[4] "전력계통해석 및 고장 복구계획", 제 41회 전력그룹 기술협력회 Workshop, 기초전력공학공동연구소, 1999. 5.
[5] M. M. Adibi, "Power System Restoration: Methodologies & Implementation Strategies", Wiley IEEE Press, Wiley IEEE Press, June, 2000.