OT-HELP 2를 사용한 조화 순차주의 검증: 역출혈 불투명성 분석에 있어서의 문제*

김선회
(중앙대학교)

Kim, Sun-Hoi. 2012. The OT-HELP 2-aided verification of Harmonic Serialism: Problems with an analysis of counter-bleeding opacity. *Studies in Phonetics, Phonology, and Morphology* 18.1, 3-26. Harmonic Serialism(HS) allows the intermediate levels of derivation within the framework of Optimality Theory (McCarthy 2008, 2010). OT-HELP 2 (Staubs *et. al*. 2010) is a software package for analyzing linguistic data within HS. In this paper, OT-HELP 2 was used to verify whether or not the HS-based analysis provides an adequate explanation for the counter-bleeding opacity. This paper investigated three types of counter-bleeding opaque phenomena occurring in the interactions between stress assignment and vowel insertion (Levantine Arabic), between tensing and consonant deletion (Korean), and between vowel lengthening and consonant deletion (English), respectively. OT-HELP 2 computed the possible patterns of forms derived from inputs with Input, Operation, and Constraint files for each type of opaque phenomenon. No result of the computations showed that HS's derivational process was able to derive the expected counter-bleeding opaque forms. According to the OT-HEP 2's computations, the determining cause of HS's inadequacy is that derivational process is always terminated with a transparent form because there is no case where an opaque form is more harmonic than a transparent counterpart in the final step of derivational process. Accordingly, it is concluded that HS has problems for solution for being verified as a general theory of phonological phenomena. **(Chung-Ang University)**

Keywords: Harmonic Serialism, counter-bleeding opacity, OT-HELP 2, gradualness, stress assignment, vowel insertion, consonant deletion, tensing, vowel lengthening

1. 서론

본고는 도출의 중간단계를 허용하는 제약 중심 이론인 조화 순차주의(Harmonic Serialism) 분석이 역출혈 불투명성(counter-bleeding opacity)을 설명할 수 있는지에 대해 논의한다. 조화 순차주의를 제안한 McCarthy (2008, 2010)는 조화 순차주의가 자연 언어에 존재할 수 없는 언어 유형을 존재 가능한 유형으로 잘못 예측하는 전통적 OT의 'too many repairs'(TMR) 문제를 해결할 수 있는 반면에(McCarthy 2008: 273), 불투명성을 설명하는 데에는 한계를 지닌다고 보았다(McCarthy 2010: 27). 본고는 제약 서열 관계의 차이에 따른 유형별 변화를 조화 순차주의적 작동 방식에 의거하여 제시하는 기능을 가지고 있는 소프트웨어 패키지인 OT-HELP를 사용하여, 조화 순차주의가 과연 그러한 한계를 지니고 있는지를 검증하고, 그러한 한계를 지닌다면 그 원인이 무엇인

* 세 분의 익명의 심사자에게 감사드린다. 투고본에 대한 이 세 분의 적절한 논평과 지적이 없었다면, 본고는 지금의 내용과 형태를 갖출 수 없었음을 밝힌다. 그러나 본고에 아직도 많이 남아있는 오류와 한계들은 전적으로 본 저자의 책임이며, 이 오류와 한계들을 극복하는 것 역시 본 저자가 앞으로 풀어야 할 숙제임을 또한 밝힌다.

지를 살펴본다. OT-HELP 2는 누구나 자유롭게 다운로드 받아 사용 가능한 OT 분석 오픈소스(open source) 소프트웨어 패키지로서 http://web.linguist.umass.edu/~OTHelp/(Staubs *et al.* 2010)로부터 다운로드 받아 사용할 수 있다.

본고는 OT-HELP 2를 사용하여 Levantine Arabic의 강세 부여와 모음 첨가의 상호작용, 국어의 경음화와 자음 탈락의 상호작용, 영어의 장음화와 자음 탈락의 상호작용에서 나타나는 불투명성에 대해 조화 순차주의 분석을 시도할 것이다. 이에 대한 분석에 앞서, 이러한 상호작용들에서 관찰되는 불투명성이 무엇인지를 가상의 예를 통해 살펴본다.

아래 (1)은 강세 부여와 모음 첨가가 상호작용하는 경우에 대한 가상의 예이다: [A]와 [E]는 강세가 부여된 모음을 대표한다.

(1) 강세 부여와 모음 첨가

a. 투명형

/pate/ → [pa.tE]　　/pakte/ → [pAk.te]　　/pakite/ → [pa.ki.tE]

b. 불투명형

/pate/ → [pa.tE]　　**/pakte/ → [pA.ki.te]**　　/pakite/ → [pa.ki.tE]

CVC 음절에 강세를 부여하되, CVC 음절이 없을 경우에는 어말에 강세가 부여되며, CVC 음절을 회피하기 위해 *i*-모음 첨가가 발생하다고 가정하자. (1a)에서는 불투명한 형태가 관찰되지 않으나, (1b)에서는 강세 부여와 모음 첨가 사이에 역출혈 관계가 형성되어 불투명형이 관찰된다: /pakte/ → [pA.ki.te]. 본고에서는 Levantine Arabic의 강세 부여와 모음 첨가의 상호작용에 대한 분석을 통해, 조화 순차주의의 도출 과정이 /pakte/ → [pA.ki.te]와 같은 불투명형을 도출할 수 없다는 것을 확인할 것이다.

아래 (2)는 경음화와 자음 탈락이 상호작용하는 경우에 대한 가상의 예이다: (2)에서 [T]는 경음화된 저해음(obstruent)을 표시한다.

(2) 경음화와 자음 탈락

a. 투명형

/palta/ → [pata]　　/pakta/ → [pata]

b. 불투명형

/palta/ → [pata]　　**/pakta/ → [paTa]**

저해음은 /k/로 대표되는 저해음 뒤에서 경음화되는 반면에, /l/로 대표되는 비저해음 뒤에서는 경음화되지 않으며(/pakta/ → [pakTa], /palta/ → [palta]), CVC 음절을 회피하기 위해 자음 연쇄의 선행 자음이 탈락한다고 가정하자. (2a)에서는 탈락하는 저해음의 영향을 받지 않아 경음화가 발생하지 않지만(/pakta/ → [pata]), (2b)에서는 경음화와 자음 탈락 사이에 역출혈 관계가 형성되어 경음화된 저해음이 나타나는 불투명형이 관찰된다(/pakta/ → [paTa]). 본고에서는 국어의 경음화와 자음 탈락의 상호작용, 영어의 장음화와 자음 탈락의 상호작용 대한 분석을 통해 조화 순차주의의 도출 과정이 이와 같은 불투명형을 도출할 수

없다는 것을 보일 것이다.

2절에서는 조화 순차주의에 대해 기술하고 OT-HELP 2의 작동 방법에 대해 설명한다. 3절에서는 (1b)의 경우에 해당되는 Levantine Arabic의 강세부여와 모음 첨가의 상호작용(Elfner 2009, McCarthy 2010)을, 4절에서는 (2b)의 경우에 해당되는 국어 경음화와 자음 탈락의 상호작용과 영어의 장모음화와 자음 탈락의 상호작용(Kenstowicz 1994)을 OT-HELP 2를 사용하여 조화 순차주의적 관점에서 분석한다. 5절에서는 본고의 요약이 제시된다.

2. 조화 순차주의와 OT-HELP 2

2.1. 조화 순차주의

본 절에서는 조화 순차주의에 대한 McCarthy(2008)의 제안을 살펴본다. 그에 따르면, 조화 순차주의는 도출의 중간 단계를 허용한다. 전체 도출 과정은 '점진적으로 조화를 증진시키는(gradual harmonic improvement)' 각 단계의 과정들로 구성된다. 입력형에 의거하여 Gen이 생성한 제한된 수의 출력 후보형들이 Eval을 구성하는 제약에 의해 평가된다. 최종 출력형에 도달하기까지의 도출 과정은 이전 단계의 평가에 의해 최적형으로 선택된 후보형이 다음 단계의 입력형이 되는 '고리순환(loop)'의 방식으로 진행된다. 그리고 각 단계에서의 출력 후보형들은 모두 그 단계의 입력형과 동일하거나 입력형에 비해 오직 한 가지의 변화만 이루어진 형태들이어야 한다.

예를 들면, 입력형 /pat/에 변화가 가해지지 않은 [pat]와 모음 *i*가 첨가된 [pa.ti]는 '단일 단계(a single level)'에서 Gen이 생성할 수 있는 출력 후보형들이다(/pat/ → [pat], [pa.ti]). 그러나 /pat/으로부터 *i*-첨가와 구개음화가 함께 발생한 [pa.ʧi]는 단일 단계에서 Gen이 생성할 수 있는 출력 후보형이 될 수 없다((/pat/ → [pa.ʧi](불가능한 도출))(McCarthy 2008, 2010). 따라서 입력형 /pat/의 최종 출력형이 [pa.ʧi]인 경우, 도출 과정은 첫 단계에서 최적형으로 선택된 pa.ti가 다음 단계의 입력형이 되어 [pa.ʧi]가 최종 출력형으로 선택되는 과정 즉, 각 단계마다 하나의 변화만 동반하는 /pat/ → pa.ti → [pa.ʧi]로 이루어져야 한다. 이러한 변화 원칙을 점진성(gradualness) 원칙이라고 한다. 점진성 원칙은 3절과 4절에서 다루게 될 현상들에 대한 조화 순차주의적 분석에서도 동일하게 적용될 것이다.

조화 순차주의의 또 다른 특징은 Eval을 구성하는 제약과 제약 서열 관계가 단계마다 다를 수 없다는 점이다. 즉, Eval은 모든 단계에서 동일한 제약과 언어 고유의 동일한 제약 서열 관계를 가지며, 이 동일한 제약 서열 관계가 각 단계의 최적형을 선택한다. 이런 방식의 도출 과정에서 선택된 최적형이 해당 단계의 입력형과 '동일한 형태가 되는(converging)' 단계를 마지막으로 도출은 종료된다. 그리고 마지막 단계에서 선택된 최적형이 최종 출력형이 된다.

McCarthy에 따르면, 조화 순차주의에서는 음운 요소의 첨가, 탈락, 음운 형태들 사이의 연결선의 첨가, 절단 등과 관련된 충실성 제약을 위반하는 작용만을 점진성과 관련된 작용으로 간주한다. 이에 의하면,

충실성 제약을 위반하지 않는 작용 즉, 음절 구조화(syllabification)와 같은 작용들은 점진성과 관련없는 작용이므로, 음절 구조화와 *i*-첨가가 단일 단계에서 발생하는 /pat/ → [pa.ti]는 점진성을 위반하지 않는다.

한편, 자음 탈락은 두 충실성 제약 MAX(place)와 MAX를 위반하는 두 가지 작용의 결과이기 때문에, 단일 단계에서 바로 자음 탈락이라는 변화가 발생할 수 없다. 예를 들면, /pakte/로부터 도출의 중간 단계를 거치지 않고 Gen에 의해 바로 [pa.te]가 될 때, 두 가지 변화 즉, MAX(place)를 위반하는 변화와 MAX를 위반하는 변화가 단일 단계에서 발생하게 된다. 이와 같은 도출은 점진성 원칙을 위반하므로, 조화 순차주의에서는 /pakte/ → [pa.te]를 /pakte/ → paK.te → pa.te → [pa.te] 도출 과정의 결과로 본다(Max(place) 위반 → MAX 위반 순, 여기서 K는 [place] 자질이 없는 자음이다).

2.2. OT-HELP 2

OT-HELP 2는 Java 기반 소프트웨어 툴로서, 조화 순차주의 방식에 의거하여 제약 서열 관계의 차이에 따른 유형별 변화를 제시하는 기능을 가진다(Mullin *et. al.* 2010: 3). 여기에서는 OT-HELP 2의 기능 및 사용 방법에 대해 상세하게 기술한 Mullin *et. al.*(2010)을 요약할 것이다.

OT-HELP 2를 작동하기 위해서는 최초 입력형이 기재된 Input 파일, Gen의 작용에 의해 생성되는 변화가 기술된 Operation 파일, 출력형 평가에 관여하는 Constraint 파일이 요구된다. 이 파일들은 'tab delimited plain text(.TXT)'로 작성되어야 하고, 확장자는 각각 .txt, .txt_OPERATIONS, .txt_CONSTRASINTS이며, 세 파일의 파일명은 반드시 동일하여야 한다. 세 파일이 하나의 폴더에 저장되어 있을 때에만 OT-HELP 2는 이 파일들을 정확히 인식한다(Mullin *et. al.* 2010: 3).

아래에는 가상의 입력형 /pate/, /pakte/, /pakite/에 앞의 (1)에서 제시된 것과 동일한 강세 부여 과정과 모음 첨가 과정이 발생하는 가상의 예를 위해 작성한 파일들이다. 이 파일들을 가지고 각 파일들의 작성 형식에 대해 기술할 것이다. 먼저, Input 파일은 다음과 같은 형식으로 작성된다.

(3) Input 파일의 예

```
[typology]
[begin tableaux]
pate        0        1
pakte       0        1
pakite      0        1
[end of tableaux]
```

[1] 자음군 단순화는 음절 구조와 깊은 관련 가진다. 예를 들어, CVC_1C_2V에서 어중(word-medial) 자음 연쇄가 단순화된 CVCV는 NoComplex, Onset, NoCoda를 모두 만족하는 CV.CV 음절 구조를 가진다. 그런데 선행 자음 C_1이 탈락하는 단순화 유형은 여러 언어에서 관찰되는 반면에, 후행 자음 C_2가 탈락하는 단순화 유형은 관찰되지 않는다(McCarthy 2008). 전통적 OT는 C_1이 탈락하는 유형뿐만 아니라, C_2가 탈락하는 유형 역시 존재 가능하다고 잘못 예측한다. 이와 같은 문제를 TMR 문제라고 한다(Pater 1999, Lombardi 2001, McCarthy 2008).

처음 두 줄은 모든 Input 파일에 존재하는 tag로서, [typology]와 [begin tableaux]라고 적는다. 다음 줄부터 기재되는 입력형은 입력형, 0, 1 형식의 세 열(column)로 구성되며, 각 열은 반드시 tab으로 구분되어야 한다. 마지막에는 [end of tableaux]라고 적는다.

Operation 파일은 Input 파일에 기재된 입력형에 가해지는 작용 또는 변화를 정의한다(Mullin *et. al.* 2010: 17). 각 작용에 대한 기술은 적어도 다섯 개의 줄로 구성되는데, 파일의 마지막에는 [end operations]라고 적는다.

(4) Operation 파일의 예

```
[operation]
[long name] InsertVowel
[active]    yes
[definition] kt         kit
[violated faith]        DEP(V)

[operation]
[long name] StressAssignment
[active]    yes
[definition] a          A
[definition] I          I
[definition] e          E
[violated faith]        DEP(Stress)
[end operations]
```

첫 줄에는 [operation]이라고 적는다. 둘째 줄에는 [long name]이라고 기재한 후, 다음 열에 작용의 명칭을 적는다. 셋째 줄의 [active]는 해당 작용의 적용 여부를 결정하는 기능을 가진다. yes가 적혀 있으면 OT-HELP 2는 해당 작용이 적용되어야 한다고 인식하는 반면에, no가 적혀 있으면 해당 작용이 적용되지 않아야 한다고 인식한다. 그러므로 no가 적혀 있을 경우에 OT-HELP 2는 해당 작용을 배제하고 yes라고 적힌 작용들만을 가지고 출력 후보형을 생성한다: Constraint 파일에 포함된 [active] 역시 동일한 기능을 가진다.

[definition]은 하나가 될 수도 있고 둘 이상이 될 수도 있다. 해당 작용에 의해 발생하는 변화가 하나라면 한 개의 정의, 둘 이상의 서로 다른 변화가 발생한다면 해당하는 만큼의 정의가 요구된다: 통합 가능한 정의들은 하나로 통합될 수 있는데, 이와 관련된 여러 가지 방식들은 Mullin *et. al.*(2010)에 자세히 제시되어 있다. 각 작용은 [definition] 줄의 중간 열에 변화되기 전 형태, 맨 마지막 열에 변화된 후의 형태를 제시하는 방식으로 정의된다. 음운 요소의 탈락이 발생하는 작용의 경우에는 마지막 열에 아무것도 적지 않는 방식으로 정의한다. (4)에서 *InsertVowel* 작용은 *i*-모음의 첨가로 인해 자음 연쇄 *kt*에 모음 *i*가 첨가되어 *kit*로 변화되는 것을 기술한 것인다. 이 작용을 정의하는데 있어서 *i*-모음이 첨가되는 환경을 설정하지 않을 경우에는 *i*-모음이 아무 곳에나 첨가되는 오류가 발생할 수 있다. 따라서 변화 환경이 기술되

어야 할 필요가 있을 경우에는 그 변화 환경을 정의에 반드시 포함하여야 한다. *StressAssignment* 작용의 경우에는 입력형에 포함된 모든 모음이 강세를 부여 받을 가능성을 가지고 있으므로, 강세 부여 환경을 설정할 필요 없이, 강세 없는 모음 [a, i, e]가 각각 강세 모음 [A, I, E]로 변화되는 방식으로 정의된다: 어느 모음이 강세를 부여 받게 되는지에 대해서는 Constraint 파일에서 정의된다.

마지막 줄에는 해당 작용으로 인해 위반되는 충실성 제약을 기재한다. *InsertVowel* 작용으로 인해 위반되는 제약이 DEP(V)므로 DEP(V)라고 적는다. Operation 파일에서 이와 같은 방식으로 충실성 제약이 기재되므로, Constraint 파일에 포함된 충실성 제약들에는 [definition] tag를 포함하지 않는다. 다만, Constraint 파일에서 해당 충실성 제약의 [active]에 no가 적혀 있거나 제약 명칭이 일치하지 않으면, 해당 작용은 작동하지 않는다.

Constraint 파일은 제약들을 정의하고 OT-HELP 2가 각 제약들을 사용할 것인지 여부를 결정한다(Mullin *et. al.* 2010: 19). 이 파일은 다음과 같은 형식으로 작성된다.

(5) Constraint 파일의 예

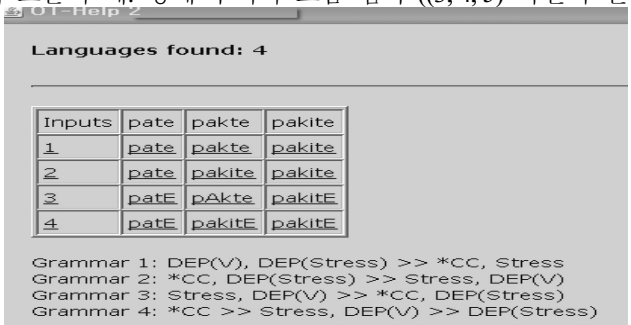
```
[constraint]
[long name]  *CC
[active]     yes
[type]       markedness
[definition] kt

[constraint]
[long name]  Stress
[active]     yes
[type]       markedness
[definition] pakte
[definition] pAtE
[definition] pakItE
[definition] p[aA]ktE
[definition] p[aA]te
[definition] p[aA]k[iI]te
[definition] pAk[Ii]tE

[constraint]
[long name]  DEP(V)
[active]     yes
[type]       faithfulness

[constraint]
[long name]  DEP(Stress)
[active]     yes
[type]       faithfulness
[end constraints]
```

첫 줄에 [constraint], 마지막 줄에 [end constrants]라고 적는다. [long name] tag에는 제약 명칭을 적는데, 앞에서 언급하였듯이, 충실성 제약의 경우 Operation 파일의 해당 작용의 [violated faith] tag에 적힌 명칭과 정확히 일치하여야 한다. [active]의 기능은 Operation 파일에서의 기능과 동일하다. [type] tag를 기재한 후 해당 제약이 유표성 제약인지, 아니면 충실성 제약인지를 반드시 적어야 한다. 유표성 제약의 정의는 [definition] tag와 함께 작성되는데, 출력형에 실현되지 않아야만 해당 유표성 제약을 충족시키는 형태를 적는 방식으로 정의한다. 예를 들면, (5)의 유표성 제약 **CC*의 정의 *kt*는 *kt* 연쇄가 실현될 경우 **CC*를 위반한다는 것을 표시한다. 유표성 제약 *Stress*의 경우 강세 부여와 관련해 *Stress*를 위반하는 형태를 정의로서 기재하는데, *pakte*는 강세 없는 [pakte], *pAtE*와 *pakItE*는 각각 [pAtE]와 [pakItE]가 실현될 경우 *Stress*를 위반하는 것을 나타낸다. 그리고 OT-HELP 2에서 *[AB]*는 'A 또는 B'의 기능을 가진다. 따라서 예를 들면, *Stress*의 정의에서 *p[aA]ktE*는 [paktE]와 [pAktE]가 실현될 경우 *Stress*를 위반하는 것을 나타낸다. 결국, 유표성 제약 *Stress*의 정의에는 CVC가 존재할 경우에는 CVC에, 그렇지 않을 경우에는 어말 CV에 강세가 부여되는 형태를 제외한 모든 형태들이 표시된다. 한편, 각 작용으로 인해 초래되는 충실성 제약이 이미 Operation 파일에 기재되어 있으므로, 충실성 제약의 정의는 별도로 기재하지 않는다.

이러한 방식으로 세 파일이 모두 작성되어 한 폴더에 저장되면, OT-HELP 2을 연 후, 주화면 하단의 <Open a File>을 클릭하여 파일을 업로드 한다. 업로드 시에는 세 파일 모두를 업로드할 필요없이, Input 파일만 업로드하면 되는데, OT-HELP 2는 Input 파일과 함께 동일 파일명을 가진 Operation 파일과 Constraint 파일을 함께 인식한다.

OT-HELP 2에 의한 도출 결과는 '도출 창(derivational window)'을 열어 확인할 수 있는데, 먼저 전체 도출 결과가 다음과 같이 제시된다.

(6) 전체 도출의 예: 강세 부여와 모음 첨가 ((3, 4, 5) 파일의 결과)



표의 첫째 줄을 제외한 각 줄은 해당 번호의 제약 서열 관계에 의해 선택된 별개의 언어 유형을 나타낸다. 즉, 1에서 제시된 강세가 부여되지 않은 출력형 [pate], [pakte], [pakite]는 Grammar 1의 DEP(V), DEP(Stress) » *CC, Stress의 결과이다. Grammar 2의 *CC, DEP(Stress) » DEP(V), Stress는 강세가 부여되지는 않았지만, /pakte/에 *i*-모음이 첨가된

[pate], [pakite], [pakite]와 같은 출력형을 도출한다. 반대로 *i*-모음 첨가가 발생하지 않고 강세가 부여된 출력형들인 [patE], [pAkte], [pakitE]는 Grammar 3의 Stress, DEP(V) » *CC, DEP(Sress)의 결과이다. 마지막으로, 4의 출력형들인 [patE], [pakitE], [pakitE]에서는 강세 부여와 모음 첨가가 모두 발생하였는데, 이는 Grammar 4의 *CC » Stress, DEP(V) » DEP (Stress)의 결과이다.

앞의 (1b)에서 강세 부여와 모음 첨가가 모두 발생하며, 그들 사이에 역출혈 관계가 형성되었을 때 예상된 불투명형은 [pAkite](/pakte/로부터)였다. 그러나 (6)에 따르면, 이러한 불투명형은 어떤 경우에도 도출되지 않는다. 이러한 불투명형을 도출하기 위해서는 /pakte/ → pAkte → [pAkite]와 같은 역출혈 도출 과정이 발생하여야 하지만, 이러한 도출 과정은 발생되지 않는다. 왜 이러한 도출 과정이 발생되지 않는가를 OT-HELP 2에서 확인하려면, /pakte/ → [pAkte]에서 도출 과정이 종료된 Grammar 3이 생성하는 /pakte/에 대한 개별 도출 과정을 살펴 보아야 한다. 개별 도출 과정을 살펴보기 위해서는 해당 도출을 클릭하면 되는데, (6)에서 셋째 줄의 셋째 열에 위치한 pAkte를 클릭하면, 다음과 같은 개별 도출 창을 통해 Grammar 3이 생성하는 /pakte/에 대한 도출 과정을 확인할 수 있다.

(7) 개별 도출의 예: /pakte/로부터 Grammar 3에 의한 도출 과정

OT-Help 2

Input: pakte

Grammar: Stress, DEP(V) >> *CC, DEP(Stress)

Step 1:

Input: pakte	Stress	DEP(V)	*CC	DEP(Stress)
pakte	-1	0	-1	0
pakite	-1	-1	0	0
☞ pAkte	0	0	-1	-1
paktE	-1	0	-1	-1

Step 2:

Input: pAkte	Stress	DEP(V)	*CC	DEP(Stress)
☞ pAkte	0	0	-1	-1
pAkite	-1	-1	0	-1
pAktE	-1	0	-1	-2

Step 1에서 CVC에 강세가 부여된 pAkte가 최적형으로 선택되어 Step 2의 입력형이 된다. Step 2에서 불투명형 [pAkite]는 *DEP(V)*뿐만 아니라 *Stress*도 위반하므로 최종 출력형으로 선택될 수 없다.

Grammar 4도 역시 **CC*가 최상위 서열에 위치하므로, *i*-모음 첨가가 먼저 발생하고, *i*-모음 첨가로 인해 CVC 음절이 존재하지 않게 되므로 강세는 어말 CV에 부여된다. /pakte/로부터 Grammar 4에 의한 도출 과

정은 다음과 같다.

(8) 개별 도출의 예: /pakte/로부터 Grammar 4에 의한 도출 과정

Input: pakte

Grammar: *CC >> Stress, DEP(V) >> DEP(Stress)

Step 1:

Input: pakte	*CC	Stress	DEP(V)	DEP(Stress)
pakte	-1	-1	0	0
☞ pakite	0	-1	-1	0
pAkte	-1	0	0	-1
paktE	-1	-1	0	-1

Step 2:

Input: pakite	*CC	Stress	DEP(V)	DEP(Stress)
pakite	0	-1	-1	0
pAkite	0	-1	-1	-1
pakIte	0	-1	-1	-1
☞ pakitE	0	0	-1	-1

Step 3:

Input: pakitE	*CC	Stress	DEP(V)	DEP(Stress)
☞ pakitE	0	0	-1	-1
pAkitE	0	-1	-1	-2
pakItE	0	-1	-1	-2

지금까지 우리는 OT-HELP 2가 어떤 방식으로 작동하는지를 살펴본 동시에, OT-HELP 2의 작동을 통한 검증으로부터 조화 순차주의가 강세 부여와 모음 첨가 사이의 상호작용에서 발생하는 불투명형을 도출할 수 없음을 알 수 있었다.

본 절에서 마지막으로 언급해야 할 점은 어떤 입력형과의 관계 하에서 충실성 제약의 위반 여부를 결정하느냐 하는 문제에 관한 것이다. McCarthy(2008)과 McCarthy(2010)에서는 이와 관련하여 서로 다른 견해가 제시되었다. McCarthy(2008: 274)는 해당 도출 단계의 입력형이 아닌 최초 입력형과의 관계 하에서 각 단계의 충실성 제약의 위반 여부가 결정된다고 제안하였는데 반하여, McCarthy(2010: 4)는 해당 단계의 입력형과의 관계 하에서 충실성 제약의 위반 여부가 결정된다고 제안하였다. 본고는 McCarthy(2008)를 따라, 충실성 제약의 위반 여부는 최초 입력형과의 관계 하에서 결정되도록 Operation 파일과 Constraint 파일을 작성하였다. 따라서 (7, 8)의 도출 과정에서 각 단계의 충실성 제약의 위반 개수는 최초 입력형과의 비교로부터 충실성 제약이 위반되는 회수를 표시한다: McCarthy(2010)를 따를지라도 본고의 분석 결과는 달라지지 않음을 확인하였다. OT-HELP 2는 그 결과를 문서 파일로 전환하여 인쇄하는 기능을 가지고 있지 않기 때문에, 프린트 스크린으로 인쇄할 수밖에 없는 한계를 가지고 있다. 3절에서는

Levantine Arabic의 강세 부여와 모음 첨가 사이의 상호작용에서 관찰되는 불투명형 분석을 통해 본 절에서 가상의 예를 통해 얻어진 결과를 다시 확인한다.

3. Levantine Arabic의 강세 부여와 모음 첨가의 상호작용

위에서 살펴본 가상의 예는 (9)의 Levantine Arabic의 강세 부여와 모음 첨가의 상호작용 현상과 거의 동일하다.

(9) Levantine Arabic (Elfner 2009: 27, McCarthy 2010: 26)
a. /katab-it-l-u/ → [ka.ta.bIt.lu] 'she wrote to him'
b. /katab-t-l-u/ → [ka.tA.bit.lu] 'I wrote to him'

(9a)는 강세가 끝에서 두 번째 CVC 음절에 부여됨을 보여 주고, (9b)는 $C_1C_2C_3$의 연쇄를 막기 위해 C_1뒤에 *i*-모음 첨가가 발생하고 있음을 보여 준다. (9b)의 강세 위치로 보아 강세 부여와 모음 첨가 사이에 역출혈 관계가 형성된다는 것을 알 수 있다. 그렇지 않다면, (9b)의 경우에 강세는 [ka.ta.bIt.lu]과 같이 되어야 할 것이다. 다음 제약들을 이 현상들에 관여하는 제약들로 가정하는데, **CC*가 **CCC*로, *Stress* 제약에서 'CVC'가 '끝에서 두 번째 CVC'로 대치된 것을 제외하곤 위의 가상의 예에 적용된 제약들과 동일하다고 보아도 무방하다.

(10) 제약
a. *CCC
CCC 연쇄에 한 개의 위반 표시를 부여함
b. Stess
강세가 없는 끝에서 두 번째 CVC에 한 개의 위반 표시를 부여함
c. DEP(V)
입력형에 상응하는 모음이 존재하지 않는 출력형 모음에 한 개의 위반 표시를 부여함
d. DEP(Stress)
입력형에 상응하는 강세가 존재하지 않는 출력형 강세에 한 개의 위반 표시를 부여함

공간상의 제약으로 /katabtlu/와 /katabitlu/를 입력형으로 작성한 Input 파일은 제시하지 않고, OT-HELP 2의 작동을 위해 필요한 Operation 파일과 Constraint 파일들을 제시하면 다음과 같다.

(11) Levantine Arabic 파일들
a. Operation 파일
[operation]
[long name] InsertVowel
[active] yes
[definition] bt bit
[violated faith] DEP(V)

```
[operation]
[long name]        StressAssignmnet
[active]   yes
[definition]       katabtlu          katAbtlu
[definition]       katabitlu katabItlu
[violated faith]   DEP(Stress)
[end operations]
```

b. Constraint 파일

```
[constraint]
[long name]        *CCC
[active]   yes
[type]     markedness
[definition]       btl

[constraint]
[long name]        Stress
[active]   yes
[type]     markedness
[definition]       katabtlu
[definition]       kat[aA]bitlu
[definition]       katAbItlu

[constraint]
[long name]        DEP(V)
[active]   yes
[type]     faithfulness

[constraint]
[long name]        DEP(Stress)
[active]   yes
[type]     faithfulness
[end constraints]
```

Input에 모음이 많이 포함되어 있기 때문에, 가상의 예의 경우와는 달리, 다른 모음에 강세가 부여되는 것을 피하기 위해 Operation 파일의 *StressAssignment* 작용에 입력형 전체를 적고 강세가 변화되기 이전과 이후의 형태를 기재하였다. 그리고 Constraint 파일에서 제약 *Stress*에는 katabtlu에 강세가 부여되지 않은 형태인 *katabtlu*, katabitlu에 강세가 CVC가 아닌 음절인 '끝에서 세 번째'에 부여된 형태와 강세가 부여되지 않은 형태인 *kat[aA]bitlu*, 강세가 두 모음에 부여된 형태 *katAbItlu*를 금하도록 정의하였다.

이 파일들과 동일한 폴더에 있는 Input 파일을 업로드하여 얻은 OT-HELP 2의 결과는 다음과 같다.

(12) *Levantine* 파일들에 의거한 OT-HELP 2의 출력형들

Languages found: 4

Inputs	katabtlu	katabitlu
1	katabtlu	katabitlu
2	katabitlu	katabitlu
3	katAbtlu	katabItlu
4	katabItlu	katabItlu

Grammar 1: DEP(V), DEP(Stress) >> *CCC, Stress
Grammar 2: *CCC, DEP(Stress) >> Stress, DEP(V)
Grammar 3: Stress, DEP(V) >> *CCC, DEP(Stress)
Grammar 4: *CCC >> Stress, DEP(V) >> DEP(Stress)

(12)에서 제시되듯이, OT-HELP 2의 작동은 (9b)의 불투명형 /katab-t-l-u/ → [ka.tA.bit.lu] 'I wrote to him'이 순차주의 도출 과정에 의해 도출될 수 없음을 보여 준다. 아래 (13)에서 제시되는 /katab-t-lu/에 대한 Grammar 3에 의한 도출 과정은 /katab-t-l-u/ → katAb-t-lu → [ka.tA.bit.lu]와 같은 역출혈 관계의 도출이 생성될 수 없는 이유를 제시한다.

(13) / katab-t-lu/로부터 Grammar 3에 의한 도출 과정

Input: katabtlu

Grammar: Stress, DEP(V) >> *CCC, DEP(Stress)

Step 1:

Input: katabtlu	Stress	DEP(V)	*CCC	DEP(Stress)
katabtlu	-1	0	-1	0
katabitlu	-1	-1	0	0
☞ katAbtlu	0	0	-1	-1

Step 2:

Input: katAbtlu	Stress	DEP(V)	*CCC	DEP(Stress)
☞ katAbtlu	0	0	-1	-1
katAbitlu	-1	-1	0	-1

Step 1에서 최상위 제약 *Stress*를 충족시키는 katAbtlu가 최적형으로 선택되어 Step 2의 입력형이 된다. Step 2에서 불투명형 katAbitlu는 Stress와 DEP(V)를 위반하여 입력형보다 더 적격의 형태가 될 수 없다.

OT-HELP 2는 조화 순차주의가 (12)에서 제시된 문법 이외에는 어떠한 문법도 생성할 수 없다고 예측한다. 예를 들어, 유표성 제약 *Stress*와 **CCC*가 동등하게 최상위에 위치하는 서열 관계 문법은 Gen에게 두 가지 변화 즉, 강세 부여와 모음 첨가를 동시에 요구하므로, 이러한 Gen의 작용은 점진성 원칙을 위배한다. 또한, 상정 가능한 서열 관계 문법 Stress » *CCC » DEP(Stress), DEP(seg)는 Step 1에서 katAbtlu를 최적

형으로 선택하지만 Step 2에서 *CCC로 인해 *i*-모음이 첨가된 katAbitlu가 상위 서열에 위치한 *Stress*를 위반하므로, Grammar 3과 동일한 결과를 가져온다. 결국, OT-HELP 2의 작동을 통한 검증에 따르면, 조화 순차주의가 Levatine Arabic에서 관찰되는 강세 부여와 모음 첨가 사이의 상호작용에서 발생하는 불투명형을 도출할 수 없다.

본고에서 지금까지 수행한 OT-HELP 2의 작동이 의존한 조화 순차주의 방식은 2.1 절에서 언급하였듯이, 충실성 제약을 위반하지 않는 작용인 음절 구조화가 점진성 원칙과 관련없는 작용이므로, 순차적으로 음절을 생성하는 작용이 발생하지 않는 방식이었다. 그러나 Elfner (2009)는 순차적으로 음절을 생성하는 방식을 사용하여 조화 순차주의가 Levatine Arabic의 불투명성을 설명할 수 있음을 보여 주었다. Elfner의 방식을 OT-HELP 2로 검증하는 것은 차후 연구로 미루지만, 아래 4절에서는 순차적 음절구조화와 관련없는 다른 불투명형을 조화 순차주의가 예측할 수 없음을 보임으로써, 조화 순차주의의 한계를 제시할 것이다.

4. 경음화/장음화와 자음 탈락의 상호작용

4.1. 국어의 경음화와 자음 탈락

국어 동사형에서 관찰되는 경음화와 자음 탈락의 상호작용에서 두 자음 연쇄의 경우에는 자음 탈락이 발생하지 않는다. 그러나 하나의 자음이 탈락되는 세 자음 연쇄의 경우, 특히 아래 (14b)에서 보듯이, /lk/, /lp/ 뒤에 저해음이 올 경우에는 어떤 자음이 탈락되느냐에 따라 투명형이 실현되기도 하고 불투명형이 실현되기도 한다. 탈락하는 자음이 어떤 것인지는 방언 차이 또는 개인 차이에 따라 다르지만(Cho 1999, Kim 2003), $C_1C_2C_3$에서 C_3는 어떤 경우에도 탈락하지 않는다. 여기에서는 /lk/ 연쇄 뒤에 /t/가 오는 동사형을 분석한다(경음은 대문자로 표시한다).

(14) 국어의 경음화와 자음 탈락의 상호작용
  a. /malta/ → [malda] ‘roll (up)’
    /makta/ → [makTa] ‘block’
  b. /malkta/ → [makTa], **[malTa]** ‘be clean’
    /palpta/ → [papTa], **[palTa]** ‘step on’

/malkta/에서 /l/이 탈락하면 투명형 [makTa]로 실현되지만, /t/가 탈락하면 경음화가 자음 탈락을 역출혈하여 불투명형 [malTa]로 실현된다.

국어의 이러한 현상을 분석하기에 앞서, 이 현상에 대한 조화 순차주의 분석의 타당성을 보다 명료하게 검증하기 위해 1절의 (2)에서 제시된 두 자음 연쇄에서 경음화와 자음 탈락이 발생하는 아래의 가상의 예부터 분석할 것이다.

(15 = (2)) 경음화와 자음 탈락
  a. 투명형

/palta/ → [pata] /pakta/ → [pata]

b. 불투명형

/palta/ → [pata] /pakta/ → [paTa]

두 자음 연쇄에서 경음화와 자음 탈락이 발생하는 경우에 관여하는 제약들로서 다음 제약들이 존재한다고 가정한다.

(16) 제약

a. *CC

[place] 자질을 가진 CC 연쇄에 한 개의 위반 표시를 부여함

b. Tensing

저해음 뒤에 오는 경음이 아닌 저해음에 한 개의 위반 표시를 부여함

b. HavePlace (McCarthy 2008:279)

[place] 자질을 가지고 있지 않은 분절음에 한 개의 위반 표시를 부여함

c. MAX

출력형에 상응하는 자음이 존재하지 않는 입력형 자음에 한 개의 위반 표시를 부여함

d. MAX(place) (McCarthy 2008:277)

출력형에 상응하는 [place] 자질이 존재하지 않는 입력형 [place] 자질에 한 개의 위반 표시를 부여함

e. IDENT(tense)

입력형의 저해음과 그에 상응하는 출력 자음의 [tense] 자질 값이 동일하지 않으면 한 개의 위반 표시를 부여함

제약 *CC는 [place] 자질을 가진 두 자음의 연쇄를 금지하는 제약이다. 아래 각주에서 언급한 대로, 이 제약을 충족시키기 위해 C_1C_2 연쇄에서 C_1이 [place] 자질과의 연결선이 끊어진다고 가정한다. 제약 Tensing을 충족시키기 위해서는 저해음 뒤에 오는 저해음이 경음화되어야 하지만, 저해음이 아닌 음의 뒤에 오는 저해음은 경음화되지 않아도 이 제약을 충족한다. HavePlace는 출력 후보형에 포함된 모든 [place] 자질을 가지고 있지 않은 자음에 의해 위반되는 제약이다. 입력형에는 존재하나 출력 후보형에 존재하지 않는 분절음의 경우에는 MAX를, [place] 자질의 경우에는 MAX(place)를 위반한다. IDENT(tense)는 입·출력 자음의 [tense] 자질 값이 동일할 것을 요구한다.

공간상의 제약때문에, 작성된 파일들을 Appendix에 제시할 것이다. 입력형 /pata/, /palta/, /pakta/에 대한 OT-HELP 2의 도출 결과는 다음과 같다.

---

[2] 사실상, C_1C_2 연쇄를 막기 위해 C_2가 아닌 C_1이 탈락하는 것은 Onset에 연결되어 있지 않은 [place] 자질을 가진 자음을 금하는 제약 CodaCond의 결과이다(McCarthy 2008). 그러나 본고의 관심이 어느 자음이 탈락하는냐 하는 것보다는 C_1이 탈락하였을 때 발생하는 C_2의 경음화에 있으므로, 여기에서는 논의의 편의를 위해 관련 제약으로서 *CC를 설정하고 *CC를 충족시키기 위해서는 C_1이 [place] 자질과의 연결선이 끊어진다고 가정할 것이다.

(17) OT-HELP 2의 출력형들: /pata/, /palta/, /pakta/

Languages found: 8

Inputs	pata	palta	pakta
1	pata	palta	pakta
2	pata	palta	pakTa
3	pata	paLta	paKta
4	pata	paLta	pata
5	pata	paLta	paKTa
6	pata	paLta	paKTa
7	pata	pata	pata
8	pata	pata	paKTa

Grammar 1: HavePlace, MAX, MAX(pl), IDENT(tense) >> *CC, Tensing
Grammar 2: HavePlace, Tensing, MAX, MAX(pl) >> *CC, IDENT(tense)
Grammar 3: *CC, MAX, IDENT(tense) >> HavePlace, Tensing, MAX(pl)
Grammar 4: *CC, IDENT(tense) >> Tensing, MAX(pl) >> MAX >> HavePlace
Grammar 5: *CC, MAX >> HavePlace, Tensing, MAX(pl) >> IDENT(tense)
Grammar 6: Tensing, MAX >> *CC, IDENT(tense) >> HavePlace, MAX(pl)
Grammar 7: *CC, IDENT(tense) >> HavePlace, Tensing, MAX(pl) >> MAX
Grammar 8: Tensing >> *CC, IDENT(tense) >> HavePlace, MAX(pl) >> MAX

(17)의 OT-HELP 2의 결과는 조화 순차주의에 따르면 경음화와 자음 탈락의 상호작용 결과로 8개의 다른 유형의 문법이 존재할 수 있음을 보여 준다. (17)에서 대문자 K와 L은 각각 [place] 자질이 없는 음 즉, 후두음 [ʔ, h]와 유사한 자질 특성을 가진 음들을 나타낸다(McCarthy 2008). 대문자 T는 경음화된 저해음을 나타낸다. 여기에서 K와 L을 구분해야 하는 이유는 /k/로부터 [place] 자질이 탈락되어 변화된 K는 [place] 자질 이외에는 저해음과 동일한 자질을 가지고 있으므로 후행 저해음의 경음화를 초래하지만, /l/로부터 [place] 자질이 탈락하여 변화된 L은 [place] 자질 없는 비저해음으로 경음화를 초래하지 않는다고 가정할 수 있기 때문이다. 따라서 K뒤에 오는 경음화된 T는 제약 *Tensing*을 위반하지 않는다.

Grammar 4와 8에 의해 도출되는 유형 4와 8을 제외한 모든 유형들은 예측 가능한 유형들이다. 즉, /pakta/에서 C_1인 /k/가 탈락하면 경음화가 발생하지 않으며, /k/가 [k, K]로 실현되는 경우에는 경음화가 발생한다. /palta/와 /pakta/로부터의 C_1의 변화 양상 역시 모두 올바르게 예측된다. 그러나 Grammar 4와 8은 자연 언어에 존재 가능하지 않은 유형을 도출한다.

먼저, Grammar 4의 *CC, IDENT(tense) » Tensing, MAX(pl) » MAX » HavePlace 서열 관계는 /palta/로부터 [place] 자질이 없는 비저해음적 자질을 가지는 L이 포함된 [paLta]를 도출한다. 그러나 /pakta/의 경우에는 /k/가 [place] 자질이 없는 저해음 [K]로 실현되지 않고, /k/가 완전히 탈

락하는 형태를 도출한다. 이러한 출력형은 다음과 같은 도출 과정의 결과이다.

(18) Grammar 4에 의한 /pakta/ → [pata]의 도출 과정

Input: pakta

Grammar: *CC, IDENT(tense) >> Tensing, MAX(pl) >> MAX >> HavePlace

Step 1:

Input: pakta	*CC	IDENT (tense)	Tensing	MAX (pl)	MAX	HavePlace
pakta	-1	0	-1	0	0	0
☞ paKta	0	0	-1	-1	0	-1
pakTa	-1	-1	0	0	0	0

Step 2:

Input: paKta	*CC	IDENT (tense)	Tensing	MAX (pl)	MAX	HavePlace
paKta	0	0	-1	-1	0	-1
☞ pata	0	0	0	-1	-1	0
paKTa	0	-1	0	-1	0	-1

Step 3:

Input: pata	*CC	IDENT (tense)	Tensing	MAX (pl)	MAX	HavePlace
☞ pata	0	0	0	-1	-1	0

이 도출 과정에서의 문제는 Step 2에서 발생한다. 동일 문법 즉, Grammar 4는 Step 2에서 *IDENT(tense)*와 *Tensing* 사이의 서열 관계와 무관하게 *CC » MAX(pl) » MAX » HavePlace에 의해 /palta/로부터 [paLta]를 도출한다. 그러나 (18)에서 제시되듯이, /pakta/로부터의 도출에서는 Step 2에서 경음화가 발생할 조건을 충족시키는 [Kt] 연쇄에 K를 탈락시키는 변화를 발생시켜 *IDENT(tense)*를 충족시킨다. 결국, 자음 탈락을 초래하는 제약들과 관계없이 경음화를 피하기 위해 자음이 탈락되는 결과가 초래된다. 경음화와 관련없는 환경 즉, [비저해음 + 저해음]에서는 자음 탈락이 발생하지 않고(/palta/ → [paLta]), 경음화와 관련된 환경인 [저해음 + 저해음]에서는 자음 탈락이 발생한다(/pakta/ → [pata]). Grammar 4가 [저해음 + 비저해음], [비저해음 + 비저해음]에서 모두 선행 자음이 [place] 자질이 없는 자음으로 변화되는 유형들을 예측한다는 점에서, [저해음 + 저해음]의 경우에만 선행 자음이 탈락되는 유형의 언어는 존재하지 않는 듯하다. 결국, 조화 순차주의 분석이 TMR 문제를 야기하는 경우라고 할 수 있다.

Grammar 8은 Grammar 4와 정반대의 유형 즉, Grammar 8에 의해 비저해음인 /l/은 탈락하지만, 저해음 /k/는 [place] 자질이 없는 [K]로 실현된다. 다시 말해서, 경음화를 발생시키기 위해 [K]가 탈락하지 않는 유형을 도출함으로써, 불투명형 [paTa]는 최종 출력형으로 실현되지 않는

다. Grammar 8에 의한 /pakta/의 상세한 도출 과정은 다음과 같다.

(19) Grammar 8에 의한 /pakta/ → [paKTa]의 도출 과정

Input: pakta

Grammar: Tensing >> *CC, IDENT(tense) >> HavePlace, MAX(pl) >> MAX

Step 1:

Input: pakta	Tensing	*CC	IDENT (tense)	HavePlace	MAX (pl)	MAX
pakta	-1	-1	0	0	0	0
paKta	-1	0	0	-1	-1	0
☞ pakTa	0	-1	-1	0	0	0

Step 2:

Input: pakTa	Tensing	*CC	IDENT (tense)	HavePlace	MAX (pl)	MAX
pakTa	0	-1	-1	0	0	0
☞ paKTa	0	0	-1	-1	-1	0

Step 3:

Input: paKTa	Tensing	*CC	IDENT (tense)	HavePlace	MAX (pl)	MAX
☞ paKTa	0	0	-1	-1	-1	0
paTa	-1	0	-1	0	-1	-1

Step 1에서 최상위 서열에 위치한 *Tensing*을 충족시키는 pakTa가 최적형으로 선택되어 Step 2의 입력형이 된다. Step 2에서는 **CC*를 충족시키는 paKTa가 최적형으로 선택된다. 마지막 Step 3에서 불투명형 [paTa]는 *HavePlace*에 관해서는 [paKTa]보다 더 적격이지만, 최상위 서열에 위치한 *Tensing*을 위반하므로 오히려 조화가 후퇴되는 결과를 초래한다: 앞에서도 언급하였듯이, 충실성 제약의 경우 그 위반 여부가 최초 입력형과의 관계에서 결정되는 방식으로 위반 회수가 표시되어 있다.

결국, 지금까지 살펴본 OT-HELP 2를 사용한 가상의 예에 대한 분석은 조화 순차주의가 두 자음 연쇄에서 경음화와 자음 탈락 사이의 상호작용의 결과로 발생하는 불투명형을 예측하지 못할 뿐만 아니라, 예상치 못한 TMR 문제를 회피할 수 없다는 것을 보여 주었다. 조화 순차주의에서 이러한 문제들이 발생하는 결정적인 이유는 불투명형이 투명형보다 더 조화를 증진시키는 형태가 아니기 때문에 언제나 투명형을 도출하는 도출 단계에서 도출이 종료되는 조화 순차주의의 본질적인 특성에 기인한다. (19)에서 Step 3에서 도출이 종료되는 것도 불투명형 [paTa]가 투명형 [paKTa]보다 더 조화가 증진된 형태가 아니기 때문이다.

조화 순차주의는 국어의 세자음 연쇄에서 경음화와 자음 탈락 사이의 상호작용의 결과로 발생하는 불투명형 역시 예측하지 못한다. 여기에서는 /malkta/에서 C_2인 /k/의 탈락과 함께 경음화가 발생하여 [malTa]가 되는 불투명형에 초점을 맞추어 분석한다. 이를 위해 (16)의 **CC* 대신 **CCC*를 설정하고 **CCC*를 충족시키기 위해서는 C_2가 탈락

하는 것으로 Constraint 파일을 작성하였다: 공간상의 제약으로 이와 관련된 Input 파일, Operation 파일, Constraint 파일의 제시는 생략하였다. 입력형 /malta/와 /malkta/에 대한 OT-HELP 2의 도출 결과는 다음과 같다.

(20) OT-HELP 2의 출력형: /malta/ 'roll (up)', /malkta/ 'be clean'

Languages found: 6

Inputs	malta	malkta
1	malta	malkta
2	malta	malKta
3	malta	malkTa
4	malta	malta
5	malta	malKTa
6	malta	malKTa

Grammar 1: HavePlace, MAX, MAX(pl), IDENT(tense) >> *CCC, Tensing
Grammar 2: *CCC, MAX, IDENT(tense) >> HavePlace, Tensing, MAX(pl)
Grammar 3: HavePlace, Tensing, MAX, MAX(pl) >> *CCC, IDENT(tense)
Grammar 4: *CCC, IDENT(tense) >> HavePlace, Tensing, MAX(pl) >> MAX
Grammar 5: *CCC, MAX >> HavePlace, Tensing, MAX(pl) >> IDENT(tense)
Grammar 6: Tensing, MAX >> *CCC, IDENT(tense) >> HavePlace, MAX(pl)

이 결과 역시 앞의 가상의 예와 동일한 문제를 가진다. Grammar 4에 의한 /malkta/ → [malta]에서 발생한 *k*-탈락은 자음 탈락에 관여하는 제약들 사이의 서열 관계와 관계없는 IDENT(tense) » Tensing에 의한 결과인데, *IDENT(tense)*를 충족시키기 위해 *tensing*을 야기하는 저해음을 탈락시킨 도출로서 TMR 문제를 야기한다. 만약 C_2가 비저해음이라면, 이 서열 관계에 의해 탈락되지 않을 것이다. 이 OT-HELP 2의 분석에 따르면, 조화 순차주의는 불투명형 [malTa]를 도출하는 어떠한 문법의 존재 가능성도 예측하지 못한다. /malkta/로부터 불투명형을 도출할 가능성이 가장 큰 Grammar 6에 의한 도출 과정은 다음과 같다.

(21) Grammar 6에 의한 /malkta/ → [maKTa]의 도출 과정

Input: malkta

Grammar: Tensing, MAX >> *CCC, IDENT(tense) >> HavePlace, MAX(pl)

Step 1:

Input: malkta	Tensing	MAX	*CCC	IDENT (tense)	HavePlace	MAX (pl)
malkta	-1	0	-1	0	0	0
malKta	-1	0	0	0	-1	-1
☞ malkTa	0	0	-1	-1	0	0

Step 2:

Input: malkTa	Tensing	MAX	*CCC	IDENT (tense)	HavePlace	MAX (pl)
malkTa	0	0	-1	-1	0	0
☞ malKTa	0	0	0	-1	-1	-1

Step 3:

Input: malKTa	Tensing	MAX	*CCC	IDENT (tense)	HavePlace	MAX (pl)
☞ malKTa	0	0	0	-1	-1	-1
malTa	-1	-1	0	-1	0	-1

(20)의 도출 과정과 마찬가지로, 이 도출 과정 역시 마지막 단계인 Step 3에서 불투명형 [malTa]는 저해음 /k/가 완전히 탈락한 결과로서 상위 서열에 위치한 *Tensing*과 *MAX*를 위반하여 다음 단계로의 도출이 차단되고 도출은 Step 3에서 종료된다. 따라서 불투명형의 도출이 불가능하다. 결국, OT-HELP 2를 사용한 분석에 따르면, 이 경우에도 조화 순차주의가 불투명형의 존재를 예측하지 못하는 한계를 지니고 있음을 알 수 있다. 4.2 절에서는 동일한 문제가 영어의 장음화와 자음 탈락의 상호작용의 결과로 나타나는 불투명형에도 발생한다는 것이 제시될 것이다.

4.2. 영어의 장음화와 자음 탈락

Kenstowicz(1994)에 따르면, 영어에서는 /nd/를 선행하는 모음은 장음화되는 반면에, /nt/를 선행하는 모음은 장음화되지 않는다. 한편, /nd/와 /nt/ 연쇄가 치경 마찰음 /s, z/를 선행할 경우, /nd/와 /nt/의 치경 폐쇄음 /t, d/는 탈락한다. 아래 (22)의 예는 이 두 음운 과정의 상호작용의 양상을 보여 준다(지금부터 단모음 ɛ를 e로 표기하고 장모음 ɛ:를 대문자 E로 표시한다. 이는 TXT 파일 작성의 편의를 고려할 뿐만 아니라 OT-HELP 2가 "ɛ:"를 하나의 분절음이 아닌 두 분절음으로 인식하는 것을 피하기 위함이다).

(22) 영어의 장음화와 자음 탈락의 상호작용(Kenstowicz 1994:72)
a. /tent/ → [tent] ‘tent’ (/tɛnt/ → [tɛnt])
/tend/ → [tEnd] ‘tend’ ((/tɛnd/ → [tɛ:nd]
b. /tentS/ → [tenS] ‘tents’ (/tɛntz/ → [tɛns])
/tendS/ → [tEnS] ‘tends’ (/tɛndz/ → [tɛ:nz])

(22a)는 /nt/ 앞에서는 모음이 장음화되지 않으나, /nd/ 앞에서는 모음이 장음화되는 것을 보여 준다. (22b)는 장음화가 자음 탈락을 역출혈하고 있음을 보여 준다. 만약 역출혈 관계에 있지 않으면, /tendZ/는 장모음이 아닌 단모음을 포함하는 형태 즉, [tenZ]로 실현되어야 한다(Kenstowicz 1994:72). 다음과 같은 제약들이 이 현상에 관여한다고 가정한다.

(23) 제약
a. *CCC
[place] 자질을 가진 CCC 연쇄에 한 개의 위반 표시를 부여함
b. Lengthening
/nd/ 앞에 오는 단모음에 한 개의 위반 표시를 부여함
c. HavePlace
[place] 자질을 가지고 있지 않은 분절음에 한 개의 위반 표시를 부여함
d. MAX
출력형에 상응하는 자음이 존재하지 않는 입력형 자음에 한 개의 위반 표시를 부여함
e. MAX(place)
출력형에 상응하는 [place] 자질이 존재하지 않는 입력형 [place] 자질에 한 개의 위반 표시를 부여함
f. IDENT(long)
입력형의 모음과 그에 상응하는 출력형의 모음의 길이가 동일하지 않으면 한 개의 위반 표시를 부여함

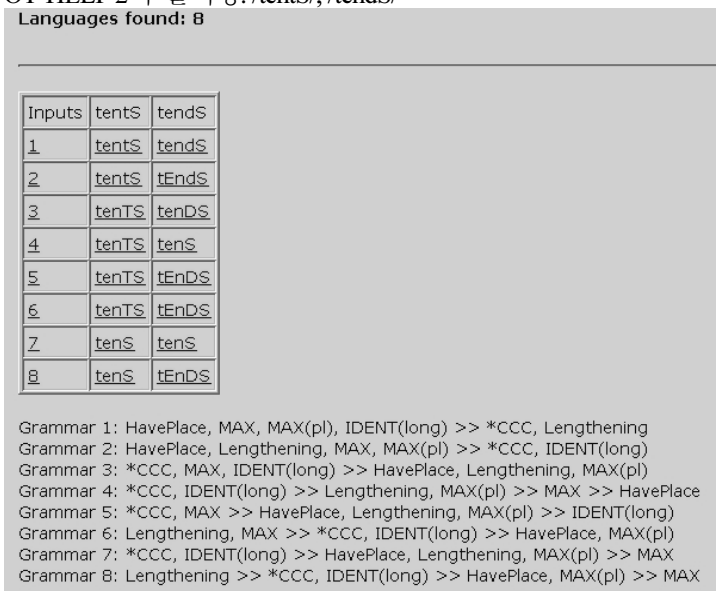
*Lengthening*과 *IDENT(long)*이 각각 *Tensing*과 *IDENT(tense)*를 대치한 것 이외에는 4.1 절에서 사용된 제약들과 동일하다. *Lengthening*은 /nd/ 앞에 위치한 모음이 장모음으로 실현될 것을 요구하는 반면에, *IDENT(long)*은 모음의 길이에 있어서 입·출력형이 동일한 값을 가질 것을 요구한다.

OT-HELP 2는 입력형 /tentS/와 /tendS/에 대해 (24)와 같은 출력형들을 도출한다.

[3] (22b)에서는 장음화와 자음 탈락의 상호작용 이외에도 무성음화와 자음 탈락의 상호작용에서도 역출혈 관계가 나타난다. 여기에서는 장음화와 자음 탈락의 상호작용에만 초점을 맞추고, 치경 마찰음 /s, z/는 모두 /S/로 표기할 것이다.

[4] 여기에서도 *CCC를 충족시키기 위해서는 C_1이 탈락한다고 가정할 것이다.

(24) OT-HELP 2의 출력형: /tentS/, /tendS/



Languages found: 8

Inputs	tentS	tendS
1	tentS	tendS
2	tentS	tEndS
3	tenTS	tenDS
4	tenTS	tenS
5	tenTS	tEnDS
6	tenTS	tEnDS
7	tenS	tenS
8	tenS	tEnDS

Grammar 1: HavePlace, MAX, MAX(pl), IDENT(long) >> *CCC, Lengthening
Grammar 2: HavePlace, Lengthening, MAX, MAX(pl) >> *CCC, IDENT(long)
Grammar 3: *CCC, MAX, IDENT(long) >> HavePlace, Lengthening, MAX(pl)
Grammar 4: *CCC, IDENT(long) >> Lengthening, MAX(pl) >> MAX >> HavePlace
Grammar 5: *CCC, MAX >> HavePlace, Lengthening, MAX(pl) >> IDENT(long)
Grammar 6: Lengthening, MAX >> *CCC, IDENT(long) >> HavePlace, MAX(pl)
Grammar 7: *CCC, IDENT(long) >> HavePlace, Lengthening, MAX(pl) >> MAX
Grammar 8: Lengthening >> *CCC, IDENT(long) >> HavePlace, MAX(pl) >> MAX

(24)에서 D와 T는 각각 [place] 자질이 없는 것을 제외하곤 [d]와 [t]와 동일한 자질들을 가진 음을 나타낸다. OT-HELP 2가 도출한 8개의 문법 유형들 가운데 Grammar 4와 Grammar 8은 4.1 절에서 제기된 것과 동일한 문제를 야기한다. 즉, Grammar 4는 장음화를 피하기 위해 /tendS/에서만 자음이 탈락하고, 장음화와 관련없는 /tentS/의 경우에는 자음이 탈락하지 않은 채, /t/가 [place] 자질이 없는 음으로 변화되는 형태가 도출된다. 이와 반대로, Grammar 8은 /tentS/에서는 /t/가 탈락되지만, /tendS/에서는 *Lengthening*을 충족시키기 위해 /d/가 탈락하지 않고 [D]로 실현된다. 결국, 불투명형 [tEnS]를 도출하는 어떠한 유형의 문법도 존재하지 않음을 예측한다.

장음화와 관련된 제약 *IDENT(long)*과 *Lengthening*의 서열 관계가 장음화가 아닌 자음 탈락에 영향을 끼치게 됨으로써 이와 같은 결과가 초래된다. 다른 환경에서는 자음의 [place] 자질이 탈락되는데, 장음화 환경에서만 모음의 장음화를 회피하기 위해 자음 전체를 탈락시키거나, 다른 환경에서는 자음 전체가 탈락되는데, 장음화 환경에서만 모음의 장음화를 위해 자음을 탈락시키지 않는 언어 유형은 존재하지 않는 듯하다. 따라서 이 역시 TMR 문제를 야기한다고 볼 수 있다.

결국, 4절의 조화 순차주의에 대한 OT-HELP 2를 사용한 검증을 통해 우리는 조화 순차주의가 역출혈 불투명형의 존재를 예측하지 못하며, 그 결정적 원인이 도출 과정이 조화를 증진시키는 형태를 도출하는 방향으로 이루어져야 한다는 조화 순차주의의 특성과는 달리 불투명형이 투명형에 비해 조화가 증진된 형태가 아니란 점에 있다는 것을 알 수 있었다.

5. 요약

본고에서는 도출의 중간단계를 허용하는 조화 순차주의 관점에서 역출혈 불투명형의 도출과 관련된 제약들 사이의 서열 관계의 차이로 인해 발생할 수 있는 언어 유형들을 OT-HELP 2를 사용하여 분석하였다. 2절에서는 조화 순차주의와 OT-HELP 2를 구성하는 중요한 특징들을 살펴보았고, 3절에서는 Levantine Arabic의 강세 부여와 모음 첨가의 상호작용에 대한 OT-HELP 2를 사용한 분석을 통해 역출혈 불투명성에 대한 조화 순차주의 분석의 문제점을 제시하였다. 4절에서는 국어의 경음화와 자음 탈락의 상호작용과 영어의 장음화와 자음 탈락의 상호작용에서 발생하는 불투명형 분석을 통해 3절의 결론을 확인하였다. 결론적으로, 본고의 OT-HELP 2를 사용한 분석은 역출혈 불투명성에 대한 조화 순차주의적 분석이 TMR 문제를 초래할 수 있으며, 불투명형을 도출하는 제약 서열 문법이 자연 언어에 존재하지 않는다는 잘못된 예측을 한다는 점을 제시하였다.

Appendix: OT-HELP 2 파일

<경음화와 자음 탈락의 상호작용: /pata/, /palta/, /pakta/>

a. Input 파일

```
[typology]
[begin tableaux]
pata        0        1
palta       0        1
pakta       0        1
[end of tableaux]
```

b. Operation 파일

```
[operation]
[long name]          PlaceDeletion
[active]     yes
[definition] kt      Kt
[definition] lt      Lt
[definition] kT      KT
[violated faith]     MAX(pl)

[operation]
[long name]          DeleteSeg
[active]     yes
[definition] [KL]
[violated faith]     MAX

[operation]
[long name]          Tensing
[active]     yes
```

[definition] kt kT
[definition] Kt KT
[violated faith] IDENT(tense)
[end operations]

c. Constraint 파일

[constraint]
[long name] *CC
[active] yes
[type] markedness
[definition] pa[kl]ta
[definition] pakTa

[constraint]
[long name] HavePlace
[active] yes
[type] markedness
[definition] [KL]

[constraint]
[long name] Tensing
[active] yes
[type] markedness
[definition] paTa
[definition] palTa
[definition] pakta
[definition] paKta

[constraint]
[long name] MAX
[active] yes
[type] faithfulness
[constraint]
[long name] MAX(pl)
[active] yes
[type] faithfulness

[constraint]
[long name] IDENT(tense)
[active] yes
[type] faithfulness
[end constraints]

REFERENCES

CHO, TAEHONG. 1999. Intra-dialectal variation in Korean cluster simplification: A stochastic approach. *UCLA Working Papers in Linguistics* 1, 1-16.

ELFNER, EMILY. 2009. Syallabification and interactions in Harmonic serialism. MS. University of Massachussetts, Amherst.

KENSTOWICZ, MICHAEL. 1994. *Phonology in Generative Grammar.* MA: Blackwell Publishing.

KIM, GYUNG-RAN. 2003. Opacity revisited: Against OT-based analysis. *Studies in Modern Grammar* 34, 171-189.

LOMBARDI, LINDA. 2001. Why place and voice are different: Constraint-specific alternations in Optimality Theory. In Linda Lombardi (ed.). *Segmental Phonology in Optimality Theory: Constraints and representations*, 13-45. Cambridge: Cambridge University Press.

MCCARTHY, JOHN. 2008. The Gradual path to cluster simplification. *Phonology* 25, 271-319.

_____. 2010. Harmonic serialism supplement to doing Optimality Theory. Ms. University of Massachussetts, Amherst.

MULLIN, KEVIN, BRIAN W. SMITH, JOE PATER, and JOHN MCCARTHY. 2010. OT-Help 2.0 User Guide. MS. University of Massachussetts, Amherst.

PATER, JOE. 1999. Austronesian nasal substitution and other NC effects. In René Kager, Joe Pater, and Wim Zonneveld (eds.). *Constraints in Phonological Acquisition*, 310-343. Cambridge: Cambridge University Press.

STAUBS, ROBERT, MICHAEL BECKER, CHRISTOPHER POTTS, PATRICK PRAAT, JOHN MCCARTHY, and JOE PATER. 2010. OT-HELP 2. Software Package. University of Massachussetts, Amherst.

Sun-Hoi Kim
Department of English Language and Literature
Chung-Ang University
221, HeukSeok-Dong, Dongjak-Gu, Seoul 156-756, Korea
E-mail: sunhoi@cau.ac.kr

received: March 21, 2012
revised: April 15, 2012
accepted: April 25, 2012