

한글 입력 방식의 구현을 위한 범용적인 복합 낱자 분석 시스템*

김 용 목 김 국†
연세대학교 언어정보학 협동과정

한글 입력 방식은 어떤 형태이건 기본 낱자들에 대한 글쇠배열과 결합 규칙이라는 두 요소를 반드시 가진다. 그런데 그 규칙을 토대로 입력 방식을 실제로 구현해 보면, 기본 낱자가 여럿 모인 복합 낱자를 입력할 때 모호성이 존재하거나 다음 글자의 연속 입력이 되지 않는 식의 문제가 발생할 수 있다. 한글은 모아쓰기라는 특성으로 인해 낱자 경계와 글자 경계라는 보이지 않는 정보가 입력 과정에서 추가로 고려되어야 하기 때문이다. 초성과 종성을 문맥에 따라 구분해야 하는 두벌식, 글쇠가 매우 적은 모바일 환경, 수십 종류의 낱자들을 조합해야 하는 옛한글이라는 조건이 더해지면 입력 방식을 기술하고 분석하는 난이도가 더욱 높아진다. 본 논문에서는 한글 입력 방식에서 쓰이는 낱자의 결합 규칙을 대결합과 소결합으로 구분해서 기술하는 체계를 제안하며, 이를 토대로 어떤 입력 방식이 주어졌을 때 복합 낱자를 입력하는 세부 규칙을 생성하고 예상되는 문제점을 자동으로 분석해 주는 도구 프로그램을 소개하였다. 그리고 모바일용 삼성 천지인과 KT나랏글 한글 입력 방식을 실제로 분석한 결과를 제시하였다.

주제어 : 한글, 옛한글, 훈민정음, 두벌식, 세벌식, 2벌식, 3벌식, 입력기, 오토마타

* 이 논문은 2017년도 정부(과학기술정보통신부, 구 미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (No.2015-0-00191, 중국어전사 한글입력기 표준개발)

† 교신저자: 김 국, 서경대학교 산업경영시스템공학과, (08826) 서울특별시 성북구 서경로 124
연구분야: 산업경영시스템공학

Tel: 02-940-7762, E-mail: kimkuk99@nare.com

서론

한글은 초·중·종성에 해당하는 세 성분이 모여서 한 글자를 형성하는 문자이다. 각 성분을 차지하는 낱자는 기본 낱자 하나로만 구성되거나, 최대 2개 또는 3개¹⁾의 기본 낱자가 결합한 복합 낱자의 형태로 존재할 수 있다. 마지막 성분인 종성은 초성과 동일한 자음 낱자를 사용해서 표현되며, 타 성분과 달리 생략도 가능하다는 차이가 있다. 한국어의 입력은 한글의 입력과 사실상 동급이며, 중국어와 일본어 같은 동아시아의 다른 언어와 달리 표기에서 한자가 차지하는 비중은 아주 작다(Ken Lunde, 2008). 그리고 문장이 아닌 글자 단위로만 짝막한 조합을 생성한다.

기계에서 한글의 입력을 위해 최광무(1978)의 두벌식 한글 입력 오토마타 연구 이후로 타자기와 컴퓨터용 글쇠배열이 먼저 연구되어 왔다(오길록 외, 1994). 2000년대부터는 연구의 흐름이 모바일 환경으로 옮겨지면서, 글쇠 수가 매우 적고 양손 대신 양 손가락을 사용하는 입력 방식이 등장했다. 처음에는 그 틀이 3×4 크기의 전화기 버튼 배열만으로 제한되어 있었으나, 스마트폰이 등장한 뒤부터는 공간과 동작의 제약이 없는 터치스크린을 기반으로 더 창의적인 입력 방식들이 개발되고 있다(강승식 외, 2016; 정 혁, 2012; 김호식 외, 2011; 임양원 외, 2010).

그런데 기기와 동작 방식을 막론하고 한글 입력 방식은 다음 두 가지 요소는 어떤 형태로든 반드시 갖추고 있다. 하나는 한 번 만에 바로 입력 가능한 기본 낱자들의 글쇠배열이며, 다른 하나는 글쇠배열에 없는 낱자를 기본 낱자들의 결합으로 생성하는 규칙이다. 새로운 한글 입력 방식을 고안할 때는 대체로 기본 낱자의 글쇠배열과 입력 방법만을 정의하며, 그로부터 파생되는 복합 낱자들은 그 낱자를 구성하는 기본 낱자들을 이어서 입력하면 자동으로 만들어진다고 간주한다. 예를 들어 ㅊ를 입력하려면 ㅊ와 ㅊ를 이어서 입력하면 되지, ㅊ가 별개의 입력 방법을 가질 필요가 없다는 것이다.

한글 입력 방식과 관련된 기존 연구 사례들은 사용자의 입장에서 더 빠르고 능률적인 글쇠배열을 고안하거나, 멀티터치와 드래그를 활용하여 더 편리한 조작법을 찾는 것 위주였다. 그러나 그런 입력 방식을 실제로 구현할 때, 개발자의 입장에서 한글 입력이라면 공통으로 거쳐야 하는 복합 낱자 분석이라는 고유한 프로세스를 자동화하는 연구는 시도된 바 없다.

어떤 한글 입력 방식으로 기본 낱자를 단독으로만 입력할 때는 보이지 않던 문제가 복합 낱자를 만들거나 여러 글자들을 이어서 입력해 보면 나타날 수 있다. 동일한 입력 순서가 두 종류 이상의 낱자로 해석 가능해서 모호성이 생길 수 있으며, 같은 자음이 종성과 다음 글자 초성 사이에 연속 입력이 되지 않고 모호성이 존재할 수 있다. 사용자는 한글을 기본 낱자들로 분해된 형태로 연속적으로 입력하지만 기계는 이 입력으로부터 원래의 글자를 조립해서 다시 복원하며, 그 과정에서 낱자와 음절 사이의 경계도 사용자가 인지하지 못하는 사이에 구분해 내야 하기

1) 옛한글까지 포함했을 때. ㅊ 같은 삼중자음과 ㅊ 같은 삼중모음이 있다.

때문이다.

그러므로 기본 낱자만을 생성하는 한글 입력 방식이 주어졌을 때, 이를 토대로 한글에서 실제로 쓰이는 모든 복합 낱자들의 생성 절차를 분석하고 논리적인 문제점을 자동으로 찾아 주는 도구가 있다면 한글 입력 방식의 구현과 검증에 도움이 될 것이다. 이 도구는 PC나 모바일용으로 현대 한글이나 옛한글 등 어떤 용도의 한글 입력 방식을 고안하더라도 공통으로 활용 가능하다.

소결합과 대결합

한글 입력 방식에는 글쇠에 없는 낱자를 입력하기 위해 글쇠에 있는 낱자들을 2타 이상 입력하여 결합하는 ‘낱자 결합’이라는 절차가 존재한다. 김용목(2012)에서도 이 개념을 제시하였는데, 이 규칙은 더 세분화해서 두 계층으로 생각할 수 있다. 하나는 ㄱ, ㄴ, ㄷ, ㅌ처럼 개념적으로 하나로 취급되는 기본 낱자를 생성하는 ‘소결합’이며, 다른 하나는 기본 낱자로부터 ㄴㅎ, ㅁㅌ, ㅊ, ㄱㅇ 같은 복합 낱자를 생성하는 ‘대결합’이다. PC와는 달리 모바일용 입력 방식은 복합 낱자뿐만 아니라 기본 낱자도 글쇠에 모두 배당되어 있지 않기 때문이다.

ㅎ이라는 자음을 예로 들면, 이것을 입력하는 방법은 입력 방식에 따라 제각각 다를 수 있다. 그러나 두벌식이라면 그 정의상 초성에서든 종성에서든 ㅎ을 입력하는 방법에는 차이가 없다. 그리고 ㄴㅎ을 입력하는 방법은 ㄴ과 ㅎ을 차례로 입력하는 것과 동일하며, ㄴㅎ의 ㅎ이든 ㄴㅎ의 ㅎ이든 ㅎ을 입력하는 방법에도 아무 차이가 없다. 그러므로 낱자 결합 규칙을 기술할 때 소결합과 대결합을 분리해서 생각하면 동일한 기본 낱자를 포함하는 복합 낱자를 다룰 때 발생하는 동일 정보의 중복을 막을 수 있다.

소결합 규칙은 ‘ㄱ+가획 → ㄱ’, ‘ㄱ+쌍자음 → ㄱㅈ’처럼 ‘초기값+과정값 → 결과값’의 형태로 연쇄적용 가능한 이항연산들로 표현된다. 입력 방식에 ‘ㅈ+쌍자음 → ㄱ’과 같은 순환이 존재한다면 이를 표현할 수도 있다.

한편, 대결합은 ‘ㄷ+ㅌ → ㅌ’, ‘ㄷ+ㅌ+ㄷ → ㅌㅌ’처럼 ‘과정값1+과정값2(+과정값3) → 결과값’의 형태로 표현된다. 대결합의 중간 과정을 구성하는 각각의 낱자들은 글쇠가 존재해서 1타 만에 단독으로 입력하거나, 혹은 소결합을 거쳐 2타 이상으로 입력하게 된다.

표 1은 삼성 천지인과 KT나랏글의 자음 낱자 결합 규칙을 소결합과 대결합의 형태로 나누어 기술한 것이다²⁾. 소결합은 초성과 종성이 한데 공유하고 입력 방식마다 규칙에 차이가 있는 반면, 대결합은 종성에만 존재하고 두 입력 방식이 서로 동일한 것을 알 수 있다.

2) 단, ‘ㄱ’에서 ‘ㄱ’처럼 이전에 거쳤던 낱자로 되돌아가는 순환 결합은 기술을 생략했다.

〈표 1〉 삼성 천지인과 KT나랏글의 자음 결합 규칙

	삼성 천지인	KT나랏글
소결합 (초·종성 공통)	ㄱ + ㄱ → ㄲ	ㄱ + 가획 → ㅋ
	ㄴ + ㄴ → ㄷ	ㄱ + 쌍자음 → ㄲ
	ㄷ + ㄷ → ㄸ	ㄴ + 가획 → ㄷ
	ㄹ + ㄹ → ㄺ	ㄷ + 가획 → ㄸ
	ㅅ + ㅅ → ㅆ	ㄷ + 쌍자음 → ㄸ
	ㅇ + ㅇ → ㅍ	ㅁ + 가획 → ㅁ
	ㅈ + ㅈ → ㅊ	ㄴ + 가획 → ㄷ
	ㅊ + ㅈ → ㅉ	ㄴ + 쌍자음 → ㄸ
	ㅋ + ㄱ → ㄲ	ㅅ + 가획 → ㅆ
	ㅌ + ㄷ → ㄸ	ㅅ + 쌍자음 → ㅆ
	ㅍ + ㅍ → ㅑ	ㅇ + 가획 → ㅎ
	ㅑ + ㅍ → ㅑ	ㅈ + 가획 → ㅊ
	ㅎ + ㅅ → ㅆ	ㅈ + 쌍자음 → ㅉ
대결합 (양 입력 방식 공통, 종성만)	ㄱ + ㅅ → ㄲ	ㄷ + ㅅ → ㄷ
	ㄴ + ㅈ → ㄷ	ㄷ + ㅌ → ㄸ
	ㄴ + ㅎ → ㄷ	ㄷ + ㅍ → ㄸ
	ㄷ + ㄱ → ㄷ	ㄷ + ㅎ → ㄸ
	ㄷ + ㅁ → ㄷ	ㄴ + ㅅ → ㄷ
	ㄷ + ㄴ → ㄷ	

두 입력 방식에서 모음의 결합 규칙을 동일한 방식으로 기술하여 대조한 결과는 표 2와 같다. 거와 ㅑ를 보면 소결합과 대결합이 각각 어떤 관점에서 낱자 결합을 기술하는지 알 수 있다. |는 삼성 천지인과 KT나랏글이 모두 글쇠를 갖고 있기 때문에 ‘ㄱ+|’, ‘-+|’, ‘ㄷ+|’는 어느 입력 방식과 결합 체계에서든 기술 형태가 동일하다.

모음의 경우 현대 한글 범위에서는 결합의 대소 구분이 자음만큼 큰 의미를 갖지는 않는다. 겹모음이 총 7개로 그리 많지 않으며, 대결합을 사용한다고 해서 규칙을 기술하는 형태가 크게 간결해지지도 않기 때문이다. 그러나 낱자의 수가 훨씬 더 많은 옛한글까지 고려한다면 모음도 기본 낱자와 복합 낱자에 대해 소결합과 대결합을 엄격하게 구분해야 결합 규칙의 전체 복잡도를 원활히 제어할 수 있다.

이렇듯 대결합과 소결합은 담당하는 역할이 서로 다르다. 대결합의 결과값은 소결합의 초기값이나 결과값에 존재하지 않아야 하며, 자신 또는 다른 대결합의 과정값에도 존재하지 않아야 한다. 예를 들어 ‘ㄱ+ㅅ → ㄲ’이라는 대결합이 존재하는데 ㅆ을 다른 낱자로 바꾸거나 ㄲ 자체를

〈표 2〉 삼성 천지인과 KT나랏글의 모음 결합 규칙

	삼성 천지인	KT나랏글
소결합 (기본 낱자)	ㅏ + ㅣ → ㅓ	
	ㅏ + ㅑ → ㅓ	
	ㅑ + ㅣ → ㅓ	ㅏ + ㅏ → ㅓ
	ㅑ + ㅑ → ㅓ	ㅏ + ㅣ → ㅓ
	ㅑ + ㅓ → ㅓ	ㅏ + 가획 → ㅓ
	ㅓ + ㅣ → ㅓ	ㅑ + ㅣ → ㅓ
	ㅓ + ㅑ → ㅓ	ㅑ + 가획 → ㅓ
	ㅓ + ㅓ → ㅓ	ㅓ + ㅣ → ㅓ
	ㅓ + ㅓ → ㅓ	ㅓ + 가획 → ㅓ
	ㅓ + ㅓ → ㅓ	ㅓ + 가획 → ㅓ
	ㅓ + ㅓ → ㅓ	
	ㅓ + ㅓ → ㅓ	
	ㅓ + ㅓ → ㅓ	
소결합 (복합 낱자)	ㅑ + ㅣ → ㅓ	ㅓ + ㅏ → ㅓ
	ㅓ + ㅑ → ㅓ	ㅑ + ㅣ → ㅓ
	ㅓ + ㅓ → ㅓ	ㅓ + ㅏ → ㅓ
	ㅓ + ㅓ → ㅓ	ㅓ + ㅣ → ㅓ
대결합 (복합 낱자, 양 입력 방식 공통)	ㅓ + ㅏ → ㅓ	ㅓ + ㅓ → ㅓ
	ㅓ + ㅓ → ㅓ	ㅓ + ㅣ → ㅓ*
	ㅓ + ㅣ → ㅓ*	ㅓ + ㅓ → ㅓ*
	ㅓ + ㅓ → ㅓ	

만드는 규칙이 소결합에 또 있어서는 안 된다는 뜻이다. 이 경우 소결합은 대결합의 부품인 ㅏ, ㅑ, ㅓ를 생성하거나 변형하는 규칙만 명시해야 한다.

소결합과 대결합의 합성

중간 과정에서 필요한 임시 낱자 생성

어떤 한글 입력 방식을 기술하는 글쇠배열과 소결합 및 대결합 규칙이 주어졌다면, 이를 토대로 분석 도구가 가장 먼저 할 일은 대결합의 각 단계에 소결합을 합성해서 복합 낱자들을 입력

하는 중간 과정을 모두 구하는 것이다.

여기서 중요한 점은 표현 형태가 생략된 중간 과정도 모두 고려해야 한다는 것이다. 이런 중간 과정은 독립된 낱자로 표현하는 대결합이 존재하지 않기 때문에 부득이하게 ‘임시 낱자’라는 형태로 표현하게 된다.

초성과 중성의 경우 임시 낱자는 자신이 향하는 대결합 결과값을 미리 표시한다. 예를 들어 KT나랏글 입력 방식에서는 ㄱ을 입력하기 위해 ‘+ㅏ’를 먼저 거치게 된다. 옛한글을 입력하는 상황이라면 ‘+ㅏ →ㅏ’라는 대결합이 정의되어 있겠지만, 현대 한글에서는 그렇지 않으므로 ‘+ㅏ’ 다음에 선택의 여지가 없이 이어질 ㄱ을 대신 표시하면 된다. 임시 낱자가 곧장 ㄱ의 역할을 하는 것이 굳이 번거롭게 ‘우ㅏ’를 거쳐서 ‘워’를 만드는 것보다 낫기 때문이다.

단, 중성의 경우는 두벌식과 세벌식이 서로 처리가 달라진다. 역시 KT나랏글을 예로 들면, ㄴ을 입력하기 위해 중간에 거치는 ‘+ㄴ’, ㅎ을 입력하기 위해 중간에 거치는 ‘+ㅇ’ 등도 모두 임시 낱자로 등록한다. 세벌식에서는 ‘+ㄴ’의 입력만으로 ㄴ을 바로 표시할 수 있고 ‘+ㅇ’이 ㅎ의 역할을 임시로 하게 할 수 있지만, 두벌식에서는 이것이 ‘인ㅏ’, ‘필요’처럼 자기 형태 그대로 쓰일 수도 있다. 그렇기 때문에 자동 완성을 할 수 없으며, 두 자음을 있는 그대로 풀어서 표시해야 한다.

연속 입력을 위해 추가로 필요한 임시 낱자 생성

두벌식 입력 방식에서는 중성이 입력된 뒤 중성이 이어지는 경우, 중성을 구성하던 낱자의 전부나 일부가 다음 글자의 초성으로 옮겨져서 그 중성과 붙는 ‘도깨비불 현상’이 발생한다. PC용 한글 입력 방식에서는 중성을 옮기는 단위를 마지막으로 입력된 자음 1타만으로 간단하게 한정해도 무방하지만, 본 논문에서 제안하는 소결합과 대결합 체계 하에서는 그 단위가 소결합 전체 또는 대결합의 마지막 과정값 전체라고 더 포괄적으로 가정하게 된다. ㅂ이나 ㅎ을 입력하는 데 2타 이상을 사용했다라도 ㅂ이나 ㅎ은 언제나 ‘+ㅂ’, ‘+ㅎ’으로 분리되어야 하기 때문이다.

이런 동작의 특성상, 두벌식에서는 고려해야 할 임시 낱자도 중성 문맥에 한해서 더 늘어난다. 지금 한 중성이 완성되었더라도 다른 낱자가 소결합 차원에서 추가로 결합 가능하다면 이것들은 모두 현재 글자의 조합으로 임시로 잡아 뒤야 한다. 그래야 나중에 도깨비불 현상이 발생할 때 대결합의 마지막 단위가 모두 뒷글자의 초성으로 넘어갈 수 있다.

그 예로는 KT나랏글을 기준으로 ‘알ㅏ’라는 어절을 입력하는 상황을 생각할 수 있다. ㄹ 다음에 ㅏ를 입력하면 ‘알ㅏ’가 된다. 현대 한글 중성에는 ㄹ 이후로 ㄹㅏ이나 ㄹㅑ, ㄹㅑ 같은 겹받침이 존재하지 않는다. 그럼에도 불구하고 한글 입력 프로그램은 ㅏ으로부터 소결합 차원에서 생성될 수 있는 ㅑ를 옆두에 두고, ‘ㄹ+ㅑ’이라는 대결합도 앞서 언급했던 ‘ㄹ+ㅇ’이나 ‘ㄹ+ㅏ’과 마찬가지로 임시로 인정해야 한다. 그렇게 하지 않으면 ‘알ㅏ’ 이후로는 조합을 강제로 끊지 않으

면 입력이 더 진행되지 않으며, ‘알자’는 연속 입력이 불가능해지기 때문이다.

표 3은 삼성 천지인과 KT나랏글이 필요로 하는 임시 낱자들을 프로그램의 분석을 통해 모두 구해서 나열한 결과이다. 겹받침들을 입력하는 중간 과정에서 반드시 필요한 임시 낱자보다는 그 뒤에 도깨비불 현상과 연속 입력의 처리를 위해 추가된 임시 낱자가 훨씬 더 많음을 알 수 있다.

또한 가획이나 쌍자음 같은 추상적인 글쇠를 쓰는 KT나랏글보다는 삼성 천지인이 임시 낱자의 사용량이 더 적다는 것도 알 수 있다. ㅎ 결합만 제외하면 삼성 천지인의 임시 낱자는 KT나랏글의 임시 낱자에 완전히 포함된다.

〈표 3〉 삼성 천지인과 KT나랏글의 임시 종성 낱자

	삼성 천지인	KT나랏글
중간 과정에 필요한 것	ㄴㅅ, ㄹㄷ, ㄹㅇ (3개)	ㄴㅅ, ㄴㅇ, ㄹㄷ, ㄹㅈ, ㄹㅇ (5개)
완성 후에 필요한 것	ㄱㅅ, ㄱㅎ, ㄴㅅ, ㄴㅈ, ㄴㅊ, ㄹㄱ, ㄹㅈ, ㄹㅊ, ㄹㅅ, ㄹㅆ, ㄹㅈ, ㄹㅊ, ㄹㅅ, ㄹㅆ, ㄹㅈ, ㄹㅆ, ㄹㅈ, ㄹㅆ (12개)	ㄱㅅ, ㄱㅈ, ㄱㅊ, ㄱㅊ, ㄴㅅ, ㄴㅈ, ㄴㅊ, ㄹㄱ, ㄹㅈ, ㄹㅊ, ㄹㅅ, ㄹㅆ, ㄹㅈ, ㄹㅆ, ㄹㅈ, ㄹㅆ (19개)
계	15	24

종성 문맥에서 초성 전용 낱자의 입력 처리

한글 자음 낱자 중에 ㅌ, ㅍ, ㅈ은 초성에만 쓰인다. 표현 범위를 옛한글로 확장하면 이들은 종성에서도 등장 가능해지지만, 그 대신 옛한글에서는 정치음과 치두음³⁾이라는 새로운 초성 전용 낱자가 등장한다. 문제는 이런 낱자를 처음 입력하는 게 아니라 앞 글자를 입력한 뒤 이어서 입력한다면, 초성이 아니라 종성 문맥에서 입력이 시작된다는 것이다.

글쇠 수가 넉넉한 PC에서는 이 낱자들을 배당할 고유한 자리가 있으므로 이들에 한해서는 현재 초성 문맥이 아니라면 조합을 종료한 뒤, 새 낱자를 다음 글자의 초성으로 보내면 된다. 즉, 세벌식의 초성 글쇠처럼 동작하면 된다. 그러나 초성 전용 낱자가 초·종성 겸용 낱자로부터 시작해서 소결합에 의해 만들어질 수도 있으므로 분석 도구는 이에 대한 대책을 제공해야 한다.

종성 문맥에서 입력된 초성 전용 낱자를 처리하는 방법은 크게 두 가지로 나뉜다. 첫째는 표시만 뒷글자의 초성으로만 할 뿐 여전히 현재 글자에 붙은 가상의 종성으로서 조합을 유지시켜

3) ㅅ, ㅆ, ㅈ, ㅊ, ㅌ, ㅍ, ㅊ, ㅆ, ㅌ, ㅍ의 10종. KT나랏글의 자음 입력 체계처럼 ㅅ, ㅆ, ㅌ에다 변형 글쇠를 결합하는 방식으로 입력 방법을 설정하기에 알맞은 형태이다.

주는 것이고, 둘째는 현재 조합을 종료하고 다음 글자의 초성으로 조합을 새로 시작하는 것이다. 표 4는 KT나랏글 기준으로 ‘간’이라는 글자를 조합하는 중에 쌍자음을 반복해서 눌렀을 때 입력 프로그램이 행할 수 있는 동작을 나타낸 것이다.

〈표 4〉 종성에서 초성 전용 날자가 결합되었을 때의 동작

처리 방법	실행 결과
현재 조합을 유지	간 → 가ㅁㅁ → 간
현재 조합을 종료	간 → 가ㅁㅁ → 가ㅁ

조합을 종료하고 종성을 다음 초성으로 옮기는 동작은 앞서 언급했던 연속 입력을 위한 임시 날자에다가도 대신 적용할 수 있다. 예를 들어 KT나랏글로 ‘알’을 입력하는 상황에서 가획 글쇠가 입력되었다면, 임시 날자까지 동원해서 ㄹ과 ㅈ을 모두 붙들고 있는 것이 아니라 ‘알’의 조합은 종료하고 초성 ㄹ만 조합하는 상태를 만드는 것이다.

이렇게 하면 계속 가획을 시도했을 때 앞 글자의 받침 ㄹ로 되돌아갈 수는 없어진다. 하지만 초성의 ㅈ과 ㄹ은 계속해서 순환할 수 있으며, ‘알’을 연속 입력하는 것 역시 가능하기 때문에 서로 일장일단이 있다. 두벌식에서도 지나치게 많은 임시 날자들을 설정하지 않고 입력에 꼭 필요한 중간 과정 날자만으로 연속 입력이라는 동일 목표를 달성할 수 있다.

모호성 해소

내부 모호성

한글 입력 방식에 정의된 대결합과 소결합 규칙을 합성해 보면 서로 다른 날자들이 동일한 입력 순서를 갖는 상황이 발생할 수 있다. 이를 감지하려면 분석 도구는 대결합에 명시된 순서대로 각 날자를 실제로 결합했을 때, 대결합의 단계가 바뀌는 곳에서 소결합도 여전히 진행 가능하지 않는지 점검해야 한다.

일례로, 표 5는 삼성 천지인의 모음 입력 체계를 확장해서 옛한글까지 입력하는 것을 가정했을 때 존재하는 모든 모호성을 프로그램의 분석으로 찾은 것이다.

‘ㄱ+ㅇ+ㅇ+ㅇ+ㅣ’의 경우 현대 한글에서는 ‘ㄱ+ㅇ’로만 해석되기 때문에 거이지만, 옛한글에서는 ‘ㄱ+ㅣ’로 해석되어 ㅅ가 될 수도 있다. 대결합으로 ‘ㄱ+ㅇ’와 ‘ㄱ+ㅣ’를 명시한 것까지는 문제가 없으나, 여기에다 각각의 홀모음을 만드는 소결합을 실제로 합성하면 모호성이 생긴다.

〈표 5〉 삼성 천지인의 모음 결합 규칙을 옛한글에 적용했을 때 발생하는 모호성

입력 순서	소결합 우선	대결합 우선
·	ㅏ	ㅑ
· ·	ㅓ	ㅕ
· · -	ㅗ	ㅛ
· -	ㅛ	ㅜ
·	ㅓ	· ㅕ
· ·	ㅕ	· ㅓ
· ·	ㅕ	· ㅓ
- ·	ㅓ	ㅛ
- ·	ㅓ	ㅛ
- · ·	ㅓ	ㅛ
- · · ·	ㅓ	ㅛ
- · · ·	ㅓ	ㅛ

KT나랏글의 소결합을 합성했을 때는 이런 모호성이 없지만, 나랏글은 ㅓ와 ㅛ, ㅏ와 ㅓ가 한글쇠에 중첩 배당되었다는 특성상 ⇒ 같은 다른 모음에서 모호성이 발생하는 것은 여전히 변함없다.

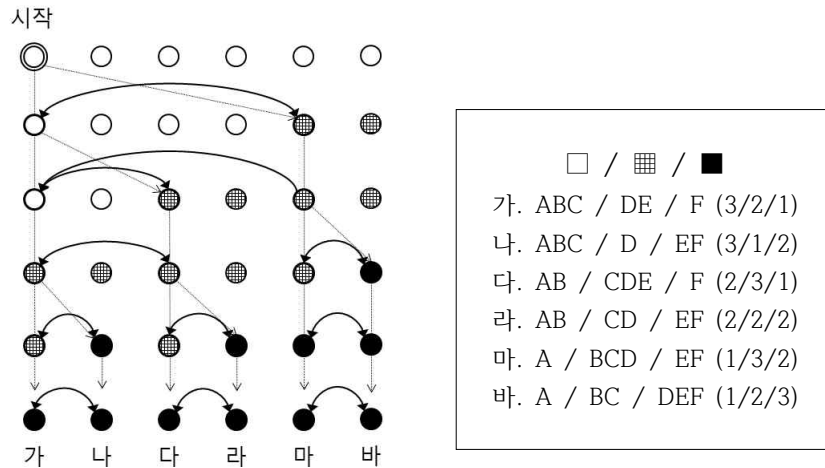
모호성을 해소하는 방법으로 본 논문에서는 ‘사전 구분’과 ‘사후 보정’이라는 두 옵션을 제안한다. 사전 구분은 한 소결합이 끝났을 때 미리 정해진 구분자를 입력해서 다음 낱자는 현재 단계의 소결합이 아니라 다음 단계의 소결합으로 강제로 가게 한다. 이를 ‘ㅓ+구분+ㅓ’로 입력하는 것과 같다.

한편, 사후 보정은 일단 낱자를 입력부터 한 뒤, 복합 낱자의 생성이 사용자의 의도대로 되지 않기 시작했을 때에 생성 방식을 변경한다. 이를 ‘ㅓ+보정+ㅓ’로 입력하는 것과 같다.

모호성 해소 기능이 동작하는 구체적인 원리는 다음과 같다. 동일한 낱자 입력에 대해 두 종류 이상의 대결합과 소결합이 존재하는 경우, 소결합을 가장 길게 적용한 것부터 순서대로 정렬하여 맨 앞에 등장하는 결합을 출발점으로 삼는다. 그리고 소결합과 대결합을 동시에 적용할 수 있는 지점에서는 가지를 분기시켜 마지막 단계에 이르기까지 binary tree를 만든다.

이런 체계에서 ‘사전 구분’은 tree의 분기점에서 새로운 가지로 가는 것과 같으며, ‘사후 보정’은 tree의 현 위치에서 가장 가까운 sibling을 교대로 왕래하는 것과 같다. 사전 구분은 새로운 낱자를 결합하지 않은 채 내부 상태만 바꾸기 때문에(ㅓ로 이어지는 ㅓ와, ㅓ로 이어지는 ㅓ가 따로) 임시 낱자라는 vertex의 개수를 늘린다. 그 반면, 사후 보정은 분기점을 지난 후에도 언제든지 사용 가능하지만 낱자 결합 규칙이라는 edge의 개수를 증가시킨다.

그림 1은 ABCDEF라는 가상의 낱자 입력이 AB, ABC, BC, BCD, CD, CDE, DE, DEF 등 3단계의 다양한 소결합으로 이뤄진 6가지 대결합으로 해석 가능한 극단적인 상황을 설정한 예이다. 이때 어떤 모양의 tree가 만들어지고 입력 중에 ‘사후 보정’을 선택하면 낱자 결합 상태가 어떻게 바뀌는지를 나타내 보았다.



(그림 1) 상태 분기를 통해 내부의 결합 모호성을 해소하는 binary tree의 예

더 짧은 소결합으로 자동 재해석

어떤 낱자가 입력되었고 다음 낱자가 현재의 낱자와 결합할 만한 것이 전혀 없다면 그 낱자는 다음 글자로 넘어간다. 그런데 현재까지 입력된 낱자 결합의 해석 방식을 바꿔서 새 낱자와 결합이 가능하다면 그 방식으로 현재 조합을 계속 유지시킬 수 있다. 단, 새로운 해석 방식은 이전의 해석 방식보다 동일 단계의 소결합 길이가 더 짧고 과정 수가 더 많아야 한다.

삼성 천지인 입력 방식에 이 조건에 부합하는 예가 존재한다. π가 소결합으로 존재하고 ㄱ가 대결합으로 존재하나, 이번에는 현대 한글만 입력하기 때문에 ㅍ는 생각하지 않기로 한다. 이때 ‘-+·+·+·+|’를 입력하면 π 이후로 |가 결합 가능하지 않기 때문에 π와 |가 그냥 별도의 낱자로 분리된다.

옛한글이었다면 ㅍ와 ㄱ의 구분을 위해 앞서 다뤘던 모호성 해소 기능이 필요하지만, 현대 한글에서는 그런 것이 필요하지 않다. 그 대신 단독 소결합 π(ABC)가 ‘T+|’(AB/CD)라는 대결합으로 나뉘면 낱자 결합이 모호성 없이 계속 진행 가능하므로 이것을 프로그램이 자동으로 분석해서 찾아 주면 된다. 첫 소결합의 길이가 π에서 T로 짧아진 대신 단계가 더 늘어난 것이 핵심이다.

이것은 일반적인 모호성 해소와는 성격이 다른 기능이지만, 입력의 편의를 위해서 모호성을 해소하는 것과 개념적으로 비슷한 방법론이 동원된다고 볼 수 있다. 현대 한글 범위에서는 ‘ㅍ+ㅣ → ㅑ’ 하나만 존재하지만, 이 역시 범위를 옛한글로 확장하면 삼성 천지인 입력 방식에는 표 6과 같은 특수한 경우가 더 존재한다.

〈표 6〉 삼성 천지인의 모음 결합 규칙을 옛한글에 적용했을 때 결합 구조가 재해석되는 상황

낱자 결합	기존 결합 구조	재해석 후
ㅍ + ㅣ → ㅑ	ㅍ + ㅣ	ㅣ ㅑ
ㅑ + ㅣ → ㅑ	ㅑ + ㅣ	ㅣ ㅑ
ㅑ + ㅡ → ㅑ	ㅑ + ㅡ	ㅑ ㅡ
ㅑ + ㅣ → ㅑ	ㅑ + ㅣ	ㅡ ㅑ
ㅑ + ㅡ → ㅑ	ㅑ + ㅡ	ㅡ ㅑ
ㅑ + ㅅ → ㅑ	ㅑ + ㅅ	ㅑ ㅅ*

마지막 예인 ‘ㅑ+ㅅ’의 경우, ㅑ는 단독 소결합인 반면 ㅑㅅ는 대결합 단위라는 것이 흥미롭다. ㅑ는 편의상 ㅍ, ㅑ, ㅑ와 마찬가지로 기본 낱자로 분류되는 편이지만, 더 분해해서 ‘ㅑ+ㅣ’로 이루어진 복합 낱자라고 볼 수도 있다. 이것은 답이 하나로 정해져 있지 않은 문제이므로, 프로그램이 대결합과 소결합의 경계를 유연하게 넘나들며 ㅑ를 ㅑㅅ와도 연결할 수 있는 것이 복합 낱자 입력 방식의 분석에 더 유리할 것이다.

ㅑ와 ㅅ가 결합하면 원래는 ㅑㅅ가 되어야 한다. 그러나 이런 낱자는 존재하지 않기 때문에 앞에서 소개한 ‘자동 완성’ 원칙에 따라 ㅑㅅ가 바로 선택되었다.

음절 경계 모호성

한 글자 내부의 낱자 결합 과정에서 발생하는 모호성은 초·중·종성 어느 낱자어나 글쇠배열의 별식과도 무관하게 존재한다. 그러나 음절 경계 모호성은 두별식 체계에서 종성과 초성 사이에 한해 존재한다.

먼저, 초성에 대결합이 존재한다면 그 결합 결과에 해당하는 자음은 내부 소결합 구조가 어떠한 원천적으로 연속 입력이 불가능하다. 종성 문맥에서 입력된 자음은 단순한 결합 불가로 인한 조합 종료, 도깨비불 현상, 초성 전용 낱자 등 그 어떤 경우에도 마지막 1타 또는 대결합의 마지막 단계 묶음에 해당하는 자음만이 뒷글자의 초성으로 넘어가기 때문이다.

이 문제를 피하기 위해 PC용 두별식 한글 입력 방식들은 쌍자음을 Shift를 동원해서 1타만으로 입력하게 해서 초성에서 대결합을 제거했다. 모바일용 입력 방식들은 초성 쌍자음을 단독 입

력할 수는 없지만 대결합이 아닌 소결합에 뒤서 종성 문맥에서도 언제나 연속 입력이 가능하게 했다.

하지만 옛한글 입력을 지원한다면 초성의 구조가 그렇게 간단할 수 없어진다. 이런 이유로 인해 두벌식 옛한글 입력 방식은 글쇠 수가 많은 PC 환경이라 해도 조합을 강제 종료해서 음절을 구분하는 글쇠가 별도로 배당되어 있다. ‘값가’와 ‘값까’(일반적인 조합 종료), ‘갑시’와 ‘가씨’(도깨비불 현상) 같은 문자열을 서로 구분해서 연속 입력을 할 수 없기 때문이다.

초성의 대결합 외에 다른 모호성으로는 종성이 소결합에 의해 2타 이상의 타수로 입력되고 그 소결합이 내부에서 두 개의 자음으로 또 쪼개지는 것이 가능한 경우이다. 이것은 삼성 천지인에서 ‘구카’와 ‘국가’, ‘장마’와 ‘자아’를 구분할 수 없는 것이 대표적인 예이며, 연속 입력이 불가능한 자음의 조합이 이 외에도 더 있다.

표 7은 연속 입력이 불가능한 모든 경우를 역시 프로그램의 도움을 받아 정리한 것이다. 삼성 천지인에는 현대 한글을 기준으로 총 76가지 경우가 존재하는데, 이는 초성과 종성의 전체 조합 가짓수인 $19 \times 28 = 532$ 와 비교했을 때 정확하게 1/7에 해당한다. 모든 종성에는 연속 입력이 불가능한 초성이 적어도 하나 이상 존재한다.

〈표 7〉 삼성 천지인에서 연속 입력을 할 수 없는 종성과 초성 조합

종성	초성	계
ㄱ, ㄲ, ㅋ / ㄹ	ㄱ, ㄲ, ㅋ	12
ㄴ, ㄴ	ㄴ, ㄴ	4
ㄷ, ㅌ / ㄹ	ㄷ, ㅌ, ㅌ	9
ㅁ, ㅇ / ㄹ	ㅁ, ㅇ	6
ㅂ, ㅃ / ㄹ, ㄹ	ㅂ, ㅃ, ㅃ	12
ㅅ, ㅆ, ㅎ / ㄴ, ㄴ, ㄹ, ㄹ, ㅅ	ㅅ, ㅆ, ㅎ	24
ㅈ, ㅉ, / ㄴ	ㅈ, ㅉ, ㅉ	9

그에 반해 KT나랏글은 음절 경계 모호성이 전혀 존재하지 않는다. 이는 단독으로 낱자를 시작할 수 없는 ‘가확’과 ‘쌍자음’이라는 글쇠만으로 소결합을 수행함으로써 중간 쪼개짐의 가능성을 제거했기 때문이다.

음절 경계 모호성은 음절 구분 글쇠를 별도로 두거나 종성의 입력이 시작된 뒤부터 조합 종료 타이머를 설정해서 해소하면 된다. 전자는 PC에서 주로 사용해 온 방식이고 후자는 모바일용 입력 방식이 주로 사용하는 방식이다. PC용 입력 방식이라도 옛한글을 입력할 때는 수동 음절 구분이 필요하며, 현대 한글의 경우 Shift를 누르지 않고 글쇠 수도 지금보다 더 늘리지 않는 입

력 방식을 추구한다면 음절 구분을 위해 다른 방법을 찾아야 한다. 후자에 대해서는 김 국 외 (2008)의 연구에서도 언급하고 있으며, 실제로 2010년에 발표된 Google 단모음 키보드는 ㄱ, ㄷ, ㅂ, ㅅ, ㅈ 계열 자음이 연결될 때에 한해 모호성이 존재한다.

허용 범위 제약 기능과의 연계

과거의 KS X 1001 한글 코드는 ‘똥’, ‘뽕’ 등 많은 글자들을 표현할 수 없는 것으로 인해 비판을 받았다. 지금은 한글 입력 프로그램이 그렇게 한글 코드 자체의 물리적 한계에 연연할 필요는 없어졌으나, 임의의 작은 한글 글자 집합에 맞춰서 동작하는 기능 자체는 여전히 특수한 용도가 있으므로 갖출 필요가 있다.

한글 허용 범위의 제약 기능이 쓰이는 대표적인 상황은 한글이 중국어나 일본어의 같은 타 언어를 발음으로 입력하는 매개 문자로 쓰일 때이다(신은주 외, 2016). 이때는 굳이 모든 한글이 입력 가능하지 않아도 된다. 예를 들어 중국어 발음이라면 ㄹ은 ‘ㄹ’을 만들 때만 종성으로 사용하고, 나머지 경우에는 일찌감치 초성으로 보내는 것이 불필요한 도깨비불 현상을 줄이고 초성 검색을 구현하는 데에 유리하다. 그런데 허용 범위 제약을 오류 없이 구현하기 위해서는 기존 한글 입력 방식의 낱자 대소 결합 구조를 분석해야 하며, 임시 낱자를 생성할 때 수립했던 것과 유사한 전략이 여기에서는 글자 단위로 적용된다.

중간 과정과 두벌식 연속 입력을 위해 필요한 글자 처리

먼저, 허용되는 글자를 입력하는 중간 과정에 거치는 글자가 허용 범위에서 누락된 경우, 이를 허용해야 한다. 일례로, KS X 1001에는 ‘뽕, 쌍, 썸, 쌍, 썸’이 들어 있지만 중간 과정인 ‘뽕, 쌍, 썸, 썸’은 존재하지 않는 것으로 잘 알려져 있다. 모바일은 단순히 받침 유무 수준을 넘어서 이런 예가 훨씬 더 많아진다. 삼성 천지인은 ‘뽕’을 입력하기 위해서 KS X 1001에 없는 ‘뽕’을 먼저 입력해야 한다. 입력 방식의 이런 특성을 면밀하게 분석하지 않고 단순히 금지 글자의 조합을 끊기만 하면 예외가 너무 많아져서 허용 범위 제약 기능을 사실상 구현할 수 없다. 이런 분석도 도구에 의한 자동화가 필요하다.

이렇게 허용 글자의 입력을 위해 부득이하게 임시로 허용하는 글자는 아무 차이 없이 원형 그대로 표시할 수 있지만, ‘ㅅㅅ’나 ‘트ㅇ’처럼 풀어 쓴 형태로 조합만 유지시켜서 표시할 수 있고, 그냥 ‘썸’의 입력만으로 ‘쌍’을 자동 완성시켜 보여줄 수도 있다. 완성 후보가 둘 이상 존재할 때는 타수가 가까운 것을 우선 선택하는 것이 바람직하다. 삼성 천지인 기준으로 ‘쌍’에 대해서는 ‘쌍(ㅅ 3연타)이 아닌 ‘쌍(ㅅ 2연타)을 먼저 제시하고, ㅅ을 실제로 3연타했을 때에 진짜

‘쌌’이 표시되게 하면 된다.

두벌식 글쇠배열에서는 중성이 있는 허용 글자를 입력한 후에 다음 글자를 연속 입력하는 것도 고려해야 한다. KT나랏글 기준으로 ㄱ이라는 접받침 이후에 ㅅ으로부터 파생 가능한 ㄱㅅ, ㄱㅆ, ㄱㅈ 같은 대결합을 임시 낱자 형태로 모두 미리 생성하는 것과 같다.

그러나 동일한 소결합 계열로 파생 가능한 글자를 모두 남겨 두면 예외 처리해야 하는 글자가 너무 많아진다. 그렇기 때문에 이런 글자는 대결합의 마지막 단계에 속하는 자음 낱자 하나만 다음 글자의 초성으로 옮기는 것이 바람직해 보인다. ‘뱌’ 다음에 가획을 누르면 곧장 ‘발교’이 되고 조합은 표에만 단독으로 존재하여 비을 왕복하는 것이다.

쓰이지 않는 낱자 결합 제거

허용되는 한글의 집합이 매우 작은 경우, 특정 낱자는 소결합이나 대결합 규칙에 지정되어 있음에도 불구하고 전혀 쓰이지 않았을 수도 있다. 외국어의 발음을 한글로 표기하기 위해 일부 옛한글 낱자를 사용하더라도 사용하는 낱자보다 사용하지 않는 낱자가 훨씬 더 많은 경우가 대부분이다.

사용되지 않는 낱자 결합 규칙을 제거하기 위해서 분석 도구가 가장 먼저 할 일은 허용 집합에 존재하는 모든 한글들의 각 성분에 대해서 한 번이라도 등장한 적이 있는 낱자는 유효하다고 표시하는 것이다. 대결합을 통해 입력되는 복합 낱자라면 그 결합의 각 단계를 구성하는 낱자들도 모두 유효하다고 간주된다.

이렇게 통계를 낸 뒤, 대결합 중에 결과값이 유효 판정을 받지 않은 것은 곧장 제거된다. 다음으로 남는 것은 소결합인데, 소결합은 소극적으로 정리하거나 적극적으로 정리할 수 있다.

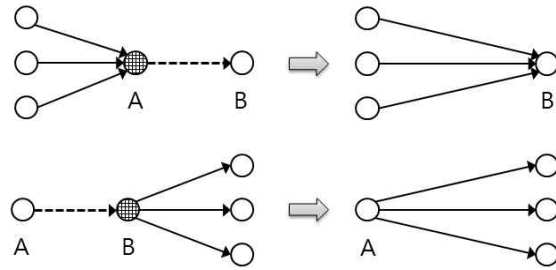
‘소극적인 정리’는 낱자들끼리의 결합을 그래프로 표현했을 때, 유효한 낱자로부터 한 번도 참조를 받지 않거나⁴⁾, 자신이 다른 유효한 낱자로 전혀 결합하지 않는 vertex에 해당하는 낱자는 고립된 것으로 간주하고 제거한다. 소극적인 정리는 기존 낱자들의 입력 방법이나 순서를 변경하지 않는다.

‘적극적인 정리’는 소극적인 정리 작업을 포함할 뿐만 아니라, 유효한 낱자의 입력 타수를 더 줄이는 변화도 수반한다. 어떤 A라는 낱자가 B로 바뀌는 결합 규칙 하나밖에 갖고 있지 않은 경우, A로 결합하는 낱자가 여럿 존재하면 A를 제거하고 그 낱자들의 결과값을 B로 바꾼다. 아니면 B가 다른 낱자로 결합하는 규칙이 여럿 존재하면 B를 제거하고 A가 그 낱자들로 결합하도록 규칙을 변형한다. A에서 B를 거치는 중간 과정을 제거하더라도 모호성이 발생하지 않기 때문이다. 단방향 linked list에서 한 node를 삭제할 때, 그 node의 이전 node가 가리키는 다음 목적지를 삭제되는 node의 목적지로 변경하는 것과 개념적으로 동일하다.

4) 여기서 ‘참조를 받는다는 것은, 다른 낱자로부터 결합의 ‘결과값’이 되는 것을 의미한다.

삼성 천지인의 경우 ㄱ → ㅋ → ㆁ → ㄱ의 순으로 자음 결합이 3단계로 순환하게 되어 있는데, 허용되는 한글 중에 ㅋ이나 ㆁ을 사용하는 글자가 전혀 없다면 이 정리 과정을 거쳐서 해당 낱자가 삭제되어서 순환이 2단계로 줄거나 순환 자체가 없어진다. 소극적인 정리로는 말단에 존재하는 ㆁ을 제거하는 것 정도가 ㄱ으로 되돌아가는 순환이 없는 경우에 한해서만 가능하다.

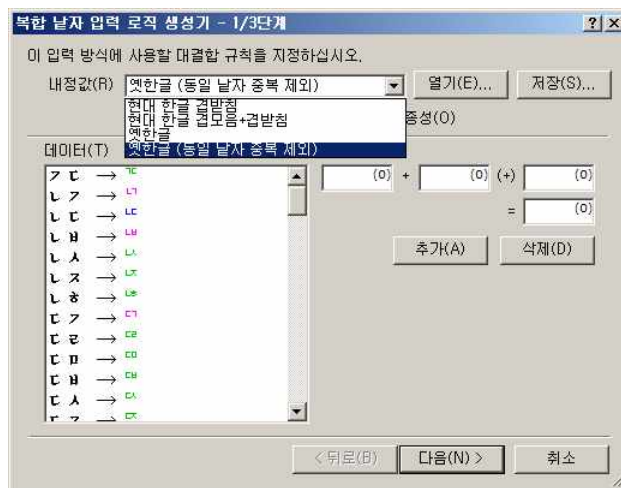
적극적인 정리의 동작 원리를 그림으로 나타내면 그림 2와 같다.



(그림 2) 낱자 결합 정리의 동작 원리

구현 결과

본 논문에서 제안하는 분석 도구는 ‘날개셋 한글 입력기’⁵⁾라는 Windows용 텍스트 에디터 겸 한글 IME에서 동작하는 부가 기능인 ‘빠른설정’의 형태로 구현되었다.

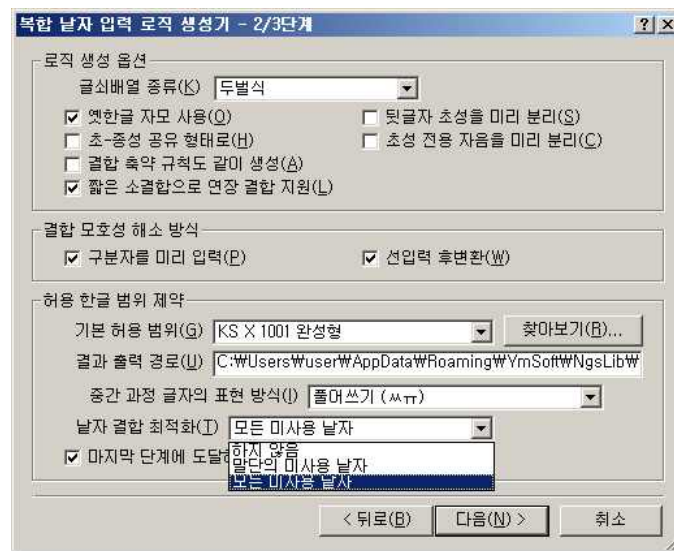


(그림 3) 분석 도구에서 대결합 규칙을 지정하는 모습

5) <http://moogi.new21.org/prg4.html>

사용자는 프로그램이 자체 제공하는 낱자 결합 규칙에서 초·중·종성별로 소결합 규칙을 지정한 뒤, 이 빠른설정에서는 대결합 규칙을 지정하면 된다. 대결합은 현대 한글 또는 옛한글의 형태로 내정값이 제공된다.

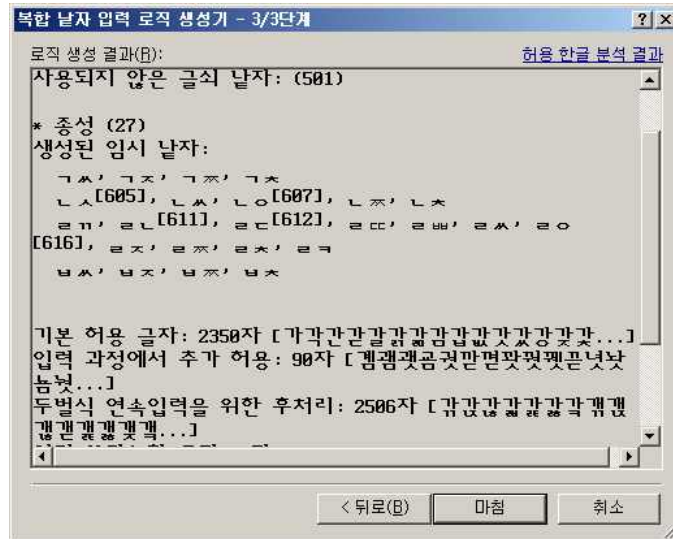
다음 단계에서는 낱자 결합 규칙을 생성하는 세부 옵션들을 지정할 수 있다. 대결합과 소결합을 합성하는 세부 방식, 모호성을 해소하는 방식, 허용 한글 범위를 제약하는 기능들을 차례로 지정한다.



(그림 4) 분석 도구에서 세부 옵션을 지정하는 모습

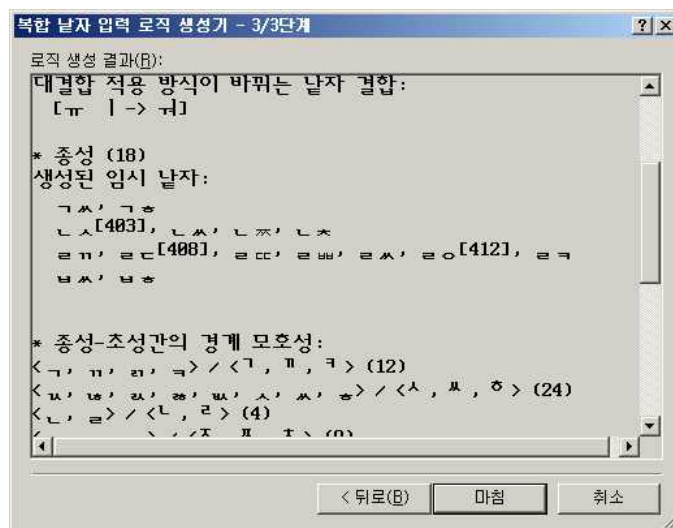
다음으로 그림 5는 KT나랏글 입력 방식의 소결합으로부터 현대 한글 복합 낱자 대결합을 합성하고, KS X 1001 완성형 한글만 입력 가능하게 제약을 적용한 결과의 일부이다. 501이라는 글쇠 낱자가 사용되지 않았다는 것은 중성에서만 쌍자음 글쇠가 사용되지 않았음을 뜻한다. 중성에서는 ㄱ으로부터 ㅂ까지 24개의 임시 낱자가 생성되었음을 알 수 있다. 그 중 연속 입력이 아니라 글자를 입력하는 과정에서 실제로 등장하는 것은 605, 607 등 숫자가 같이 표기되어 있는 5개뿐이다.

그리고 완성형 한글을 입력하는 과정에서 입력을 추가로 허용해야 하는 글자는 90자이지만, 연속 입력을 위해 중성의 마지막 결합 단계를 다음 글자로 보내야 하는 글자는 2350자보다도 더 많음을 알 수 있다. 생성된 전체 데이터는 화면 우측 상단의 '허용 한글 분석 결과' 링크를 클릭하면 볼 수 있다.



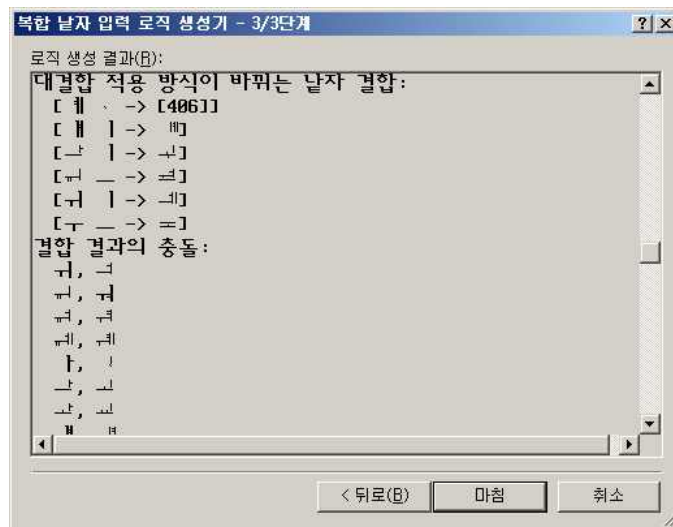
(그림 5) KT나랏글 방식에 현대 한글 KS X 1001 제약을 적용하여 복합 낱자 입력 구조를 분석한 결과

그림 6은 삼성 천지인 입력 방식의 소결합으로부터 현대 한글 복합 낱자 대결합을 합성한 결과의 일부이다. 'ㅍ+ㅣ'가 거로 바뀌는 것이 특수하게 처리되었다는 메시지가 있으며, 아래에는 음절 경계 모호성이 존재하는 종성과 초성의 쌍이 모두 출력되었다. 삼성 천지인은 종성 임시 낱자의 수가 KT나랏글보다 적다.



(그림 6) 삼성 천지인에 현대 한글 복합 낱자 구조를 분석한 결과

끝으로, 삼성 천지인 입력 방식에 옛한글 대결합을 합성하면 그림 7과 같이 훨씬 더 복잡한 결과가 도출된다. 모음에서 아래와 같이 충돌이 발생할 뿐만 아니라, 자음도 현대 한글 결합침을 기준으로 최대한 충돌을 피하게 중첩 배당된 것이 옛한글에서는 여전히 충돌을 일으킨다는 것을 프로그램이 알려 준다. ㄴㄹ, ㄷㅌ, ㅂㅍ, ㅅㅎ 같은 것이 모두 복합 낱자로 존재하기 때문이다.



(그림 7) 삼성 천지인에 옛한글 복합 낱자 구조를 분석한 결과

결 론

지금까지 본 논문에서는 기본 낱자에 대한 글쇠배열과 결합 규칙만 정의되어 있는 한글 입력 방식이 복합 낱자의 입력과 원활한 연속 입력을 지원하기 위해 필요한 세부 절차들을 살펴보았다. 그리고 입력 방식이 잠재적으로 가질 수 있는 문제점을 발견하고 해결하는 방법도 제안하였으며, 이 모든 작업을 실제로 수행하는 분석 도구를 구현하여 삼성 천지인과 KT나랏글이라는 잘 알려진 모바일 입력 방식을 분석한 결과를 제시했다.

두 입력 방식은 현대 한글 범위에서는 거에 특이사항이 공통으로 존재한다. KT나랏글에서 'ㄷ+ㅌ'만으로 곧바로 거가 되고, 삼성 천지인에서는 'ㄷ+ㅌ'가 곧바로 거가 되는데, 이것은 대결합과 소결합의 기계적인 합성만으로는 설명되지 않는다. 그 대신 각각 '임시 낱자 자동 완성'과 '더 짧은 소결합으로 자동 재해석'이라는 개념으로 일반적인 설명이 가능함을 보였다.

더 나아가 각 입력 방식들이 내부적으로 임시 조합들을 얼마나 사용하는지 확인했으며, 삼성

천지인이 음절 경계에서 모호성을 일으키는 모든 경우를 살펴보았다. 삼성 천지인은 모음의 결합 규칙이 독특하기 때문에 이 체계를 옛한글로 확장할 때 발생할 수 있는 내부 모호성과 ‘더 짧은 소결합으로 자동 재해석’ 가능성을 특별히 서로 연계해서 확인했다.

낱자 규칙을 대결합과 소결합으로 나눠서 기술한 것은 도깨비불 현상처럼 두벌식 입력 방식의 고유한 낱자 처리 동작을 명시하는 데도 도움이 됨을 알 수 있었다. 같은 범위의 한글을 입력한다면 두벌식 입력 방식은 세벌식보다 글쇠 수가 적은 대신, 글자를 완성한 뒤에도 연속 입력을 위한 임시 낱자가 추가로 필요하며 음절 경계 모호성이나 중성 문맥에서 초성 전용 낱자의 입력에 대한 대비도 필요하다. 또한, 두벌식에서는 ‘앉 → 알자’나 ‘간 → 가디’처럼 모음 없이 자음끼리만 음절 경계에서 분리되는 동작도 여러 분야에서 필요하다는 것을 알 수 있었다.

2010년대부터는 Windows와 macOS 등 운영체제의 기본 글꼴에도 KS X 1026-1⁶⁾을 준수하는 옛한글 낱자가 모아쓰기 형태로 지원되기 시작하여 옛한글에 대한 접근성이 예전에 비해 크게 개선되었다. 또한 스마트폰에서도 PC와 기술적으로 아무 차이가 없는 다국어 입출력 처리가 지원된다. 그런 만큼 한글 입력이라는 분야에서는 언어 데이터나 타자 행동 데이터를 기반으로 한 성능 개선뿐만 아니라, 복합 낱자의 결합 방식처럼 어떤 입력 방식이라도 저변에서 공통으로 의존하는 기본 기능에 대한 연구가 진행될 여지가 남아 있다. 별수와 글쇠 수는 적는데 입력 가능해야 하는 낱자는 많은 입력 방식을 설계할수록 복합 낱자의 결합을 자동으로 구현하고 분석하는 도구의 유용성은 더 커질 것이다.

참고문헌

- 장승식, 최윤승 (2016). “시각 장애인의 입력 편의성 향상을 위한 손가락 터치 기반의 한글 입력 인터페이스”, **정보과학회논문지**, 43(11), 1307-1314.
- 김 국, 유영관 (2008). “사용빈도와 표준정합성을 고려한 컴퓨터 한글자판의 개선에 관한 연구”, **대한인간공학회지**, 27(3), 7-14.
- 김용목 (2012). “**한글 입력·편집기의 통합적 설계와 구현에 관한 연구**”. 연세대학교 석사학위논문.
- 김용목, 김 국 (2016). “두벌식 한글 입력 방식의 정형적인 기술과 분석”. **제 28회 한글 및 한국어 정보처리 학술대회 발표논문집**, 27-32.
- 김호식, 전재웅, 최윤철 (2011). “터치스크린 기반 스마트폰에서의 한글 입력 기법”. **한국 컴퓨터 종합 학술대회 논문집**, 38(1), 263-265.

6) <http://www.unicode.org/L2/L2008/08225-n3422.pdf>

- 남경완 (2008). “훈민정음의 제자 원리를 통해서 살펴본 핸드폰 한글입력 방식에 대한 연구”. **한국어학**, 제41권, 325-255.
- 신은주, 이다솜, 임순범 (2016). “한글 입력 키보드를 이용한 훈민정음 기반 중국어 및 외국어 입력 방법”, **한국전자출판연구**, 제6호, 7-15.
- 오길록, 최기선, 박세영 (1994). “**한글공학**”, 대영사, 81-108.
- 임양원, 임한규 (2010). “모음의 방향성을 이용한 한글 입력 방식에 관한 연구”. **한국 멀티미디어 학회 학술대회 논문집**, 13(2), 19-22.
- 정 혁 (2012). “자모 변환 운지 거리를 최소화한 한글 자모 입력 방법의 제안”, **한국HCI학회 학술대회 논문집**, 41-43.
- 최광무 (1978). “**한글 모아쓰기에 관한 연구**”. KAIST 석사학위논문.
- Ken Lunde (2008). “CJKV Information Processing, Second Edition”, O'Reilly Media, 350-361.

1차 원고 접수: 2017. 05. 03
1차 심사 완료: 2017. 07. 05
2차 원고 접수: 2017. 09. 12
2차 심사 완료: 2017. 10. 25
최종 게재 확정: 2017. 10. 27

(Abstract)

Logic Analyzer of Composite Hangul Units for Implementation of Input Methods

Yongmook Kim

Kuk Kim

Yonsei University

A Hangul input method consists of two core components: key layout for primitive units (Natja), and binding rules to form more complex, composite units from primitive ones. However, careless design and implementation of an input method may result in ambiguity and inability of contiguous typing, when it comes to handle composite Hangul units. This is due to the collective nature of Hangul, namely, existence of implicit boundaries between units and syllables. This issue gets more challenging for the context-dependent 2-beolsik method, mobile environment having significantly fewer keys, and yet addition of the support for Old Hangul. For the sake of the management of complexity, we devised a system in which the binding rule is divided into two layers rather than one. Based on this concept, we implemented a tool program that, given the primitive units and their binding rules, generates input details of every composite unit and detects any potential ambiguity and conflicts. We presented the actual analysis result for Samsung Cheonji-in and KT Naratgeul, two of the most popular mobile Hangul input methods used in Korea.

Key words : Hangul, Hunminjeongeum, IME, 2-beolsik, 3-beolsik, Key Layout, Keyboard, Input Method, Automata