국어 연구와 전산기

김 정 수
(한양대 교수, 국어학)

1

내가 전산기를 써서 언어 자료를 처리할 마음을 먹고 준비하기 시작한 것은 1980년 7월의 일이다. 이맘때 나는 17세기 한국말의 굴곡법에 관한 박사 학위 논문을 쓰려고 종이 카드로 자료를 만들고 있었는데, 논문 마감은 겨우 3년 정도 밖에 남지 않았었다. 그 때 훑어 본 17세기 문헌은 예닐곱 가지에 불과한데, 보아야 할 문헌은 열 여섯 가지나 되었다. 시간에 쫓기지 않을 수 없었고, 그대로 진행해서 논문을 낼 자신이 없었다. 기초 문헌에서 예문을 뽑아 가며 만드는 종이 카드는 아무리 요령이 좋아도 두어 가지 이상의 용도로는 활용하기 어렵고, 학술적으로 많다고 할 수도 없는 옛글 문헌을 철저하게 요리해 낼 도구도 못 된다.

그래서 전산기를 이용해 보려고 한국 과학 기술 정보 센터[KORSTIC]라는 데를 찾아가 혼자 설계한 언어 자료의 처리 방안과 자료 문헌의 분량을 전문가에게 대략적으로 제시하면서 상담했더니, 전산 용역 회사에 일을 맡길 경우에 기계 사용료와 입력 인건비 등만 적어도 300만 원이 들겠고, 입력 설비를 직접 갖추자면 1,200만원이 들겠다고 했다. 착수에 불과한 과정을 위해서 이런 돈을 들인다는 것은 불합리했다. 그래서 단념하고 있다가 그 이듬해 6월에 알아 보니 개인용 전산기로 삼보 전자 엔지니

어링이라는 자그마한 회사에서 만든 것(SE-8001)이 있었다. 이것은 개인용 전산기로서는 최초의 국산품이라 했다. 최소한의 규모로 사는 데도 120만원이나 들었다. 막다른 골목에 몰린 것만 같은 처지라 다른 선택의 여지가 없었다. 글자판의 한글 배열도 제멋대로이고 요즘처럼 모아쓰기도 되지 않는 데다가 옛글은 더군다나 들어 있지도 않은 이 기계를 가지고 씨름하면서 용도에 맞게 개조하고 전산 언어를 배우며 길들이는 일에 다시 이태가 지나갔다.

이 기계를 옛말 연구에 활용하기 위해서 한글 기울여 풀어쓰기를 궁리해내고, 언어 처리를 할 모든 준비를 마친 때가 1983년 봄이었다. 이 때부터 부리나케 옛글의 본문을 분석적으로 입력하면서 아울러 처리하는 프로그램을 베이직(BASIC) 언어로 이리저리 짜면서 결과를 얻어 그 해 가을에 논문 한 편을 완성할 수 있었다. 그러니까 17세기 문헌 열 여섯 가지를 입력하고 처리하는 일에 서너 달 가량이 걸린 셈이다. 이렇게 해서 나온 논문이 졸속한 것임에는 틀림없지만, 자료 처리의 과정만은 지금 사정에 비추어 보아도 쾌속한 것이었다.

2

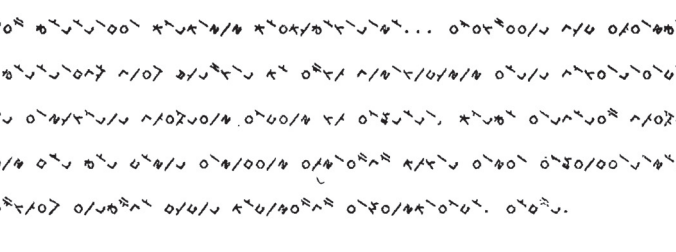
이런 면에서 풀어쓰기의 효용은 절대적인 것이었다. 얼마 전에 마침 이익섭 교수님이 한글의 모아쓰기와 풀어쓰기를 비교한 일(「국어생활」, 3호: "한글의 모아쓰기 방식의 표의성에 대하여")이 있어서 이에 대해 언급할 필요성을 느낀다.

한글 풀어쓰기는 다 아다시피 주 시경 스승께서 국문 연구소의 위원으로 활약하면서 1908년에 낸 "국문 연구안"을 통해 처음으로 주장하고, 몸소 한국말을 가르친 강습소의 수료증을 풀어 쓴 한글만으로 만들어 실천했다. 이 때부터 오늘날까지 김 두봉(1922), 최 현배(1922), 조선어 학회(1936), 조 병희(1946), 도 명보(1946), 장 봉선(1946?), 윤 덕중(1969), 양 제철(1974), 위 성인(1974, 1984) 등의 많은 사람들과 기관에서 서로 다른 한

글 풀어쓰기를 제안해 왔다.

이들 가운데 위 성인(1984)이 엽서를 통해서 한글 학회에 제안한 이른바 "한글 비껴쓰기"와 이 사람(1982)의 "한글 기울여 풀어쓰기"는 우연한 일치로 기본적인 착상이 똑같다. 위에 든 여러 사람의 풀어쓰기와 이 두 가지 풀어쓰기는 전혀 이질적이라는 것을 사람들이 모르고 있다. 앞것은 모아쓰기를 완전히 벗어난 풀어쓰기요, 뒷것은 모아쓰기의 음절 모양과 특성을 완전히 보전할 수 있게 한 풀어쓰기라는 점이 중요하다. 바꿔 말하면, 앞것은 현행 한글 맞춤법을 아예 무시하는 데 반해서, 뒷것 특히 이 사람의 방안은 맞춤법의 변경이 조금도 필요하지 않다는 점이 중요하다.

그러므로 이 익섭 교수님이 모아쓰기의 음절 모양을 파괴하는 풀어쓰기 방식의 본보기만 가지고 음절 단위로 눈에 익은 낱말이나 어형의 시각적인 모양이 파괴된다는 이유를 들어 풀어쓰기를 배척하고 모아쓰기 옹호론의 주요한 근거로 삼는 것은 이제는 타당하지 않다. 한글 기울여 풀어쓰기가 모아쓰기와 얼마나 가까운지를 다음의 보기로 확인해 주기 바란다.



이제 풀어쓰기와 모아쓰기의 시각적인 차이는 글자가 음절 단위로 왼쪽으로 기울었느냐 곧추 서 있느냐 하는 것 한 가지 밖에 남지 않은 셈이다. 바꿔 말해서 사람이 지면을 오른쪽으로 조금 기울여 보면서 차츰 길들여야 하는 것 밖에는 풀어쓰기에 대해 비판할 것이 없고, 그만한 부담에 대한 보상은 기계화에 관한 모아쓰기의 모든 장애를 완전히 해소해 주는 것이니, 비교할 바에는 저런 부담과 이런 보상 효과 밖에 비교할 거리가 없을 것이다.

이러한 기울여 풀어쓰기의 장점이 어떠하든지, 모아쓰기는 모아쓰기 대

로 편의와 특성이 있는 것인 만큼 그대로 보전하고 이용하면서 보충적으로 풀어쓰기를 이용하는 것이 슬기로운 처사라고 생각된다. 이에 대해서는 먼저 글자 생활의 일반적인 보수성이 얼마나 완강한 것인지를 이해해야 한다. 그러므로 우리는 한글의 모아쓰기와 풀어쓰기에 대해서 둘에서 하나 버리기로 대들 것이 아니라 없어서 아쉬웠던 날개 하나를 더 달아 주는 일로 여겨야 할 것이라고 말하고 싶다.

로마자의 글씨는 자그마치 2,000가지가 넘고 우리가 인쇄소의 활자 본 같은 데서 볼 수 있는 것만도 수십 가지나 되는데, 이에 비하면 너무도 빈약하기만 한 한글의 글씨 가운데 이탤릭체의 반대 방향으로 기울이고 풀어쓰는 글씨 하나를 보태자는 데 대해서는 냉담한 의견이 어찌나 많은지 안타깝다. 하물며 그것이 이탤릭체와는 비교할 수도 없이 엄청난 편익을 주는 것이오, 또한 모아쓰기와 조화하고 공존할 수 있는 완전한 조건을 갖춘 것임에랴?

3

전산기는 특히 형태론에 아주 적합하고 유용한 기계라고 생각된다. 왜냐 하면, 형태론적인 분석의 과정이 그다지 높지 않은 수준의 기계적인 처리로 감당할 만 하고, 섭렵해야 할 자료의 분량이 사람의 손과 머리만으로는 벅차지만 기계로는 문제가 되지 않고 또한 개인용 전산기의 기억 한계를 크게 벗어날 정도는 아니니 말이다.

전산기로 언어 자료를 처리하는 과정은 대개 자료의 (1) 입력, (2) 처리, (3) 인쇄, (4) 보존의 네 단계로 나눌 수 있다. 이 가운데 세째와 네째는 전산 분야에서 배울 수 있는 것이고, 앞의 두 단계는 언어 연구자가 스스로 연구해 가며 개발해야 하는 것이다. 그러므로 앞의 두 단계에 대해서만 간략하게 설명하기로 하겠다.

(1) 입력

요즘 언어의 자동 번역을 위해서는 전산기 자체가 언어를 분석하고 이해할 수 있도록 말본과 낱말 모음 등을 따로 품고 있고, 그 바탕 위에 언어 자료를 통째로 입력하도록 되어 있는 것 같다. 그러나 우리네 형편은 그런 수준에 이르지 못해서 언어 자료를 자연 상태로 입력할 수가 없다.

우리가 현실적으로 택할 만한 입력 방법은 언어 자료를 미리 분석한 다음에 분석된 단위에다가 낱낱이 패를 달아 주는 것이다. 그리해 놓으면, 그 패를 따라 찾아내기와 가르기를 할 수 있기 때문이다. 그러므로 입력을 위해서 먼저 정해야 할 것은 언어 자료를 어떠한 정도로 잘게 또는 크게 분석할 것인가 하는 것과 분석된 단위의 유형에 대해 어떤 패를 붙일 것인가 하는 것이다.

언어 분석의 깊이는 연구의 목적에 따라 달리 잡힐 것이다. 그러나 입력을 위한 분석의 깊이는 특히 한국말에서 형태론을 중심으로 삼아 어느 정도 표준적인 한계를 잡을 수 있다고 생각된다. 음소나 음절의 분석은 입력 단계에서 대비하지 않아도 자동적으로 시행할 수 있고, 월 짜임새의 분석은 형태론적인 단위에 초연할 수 없으니 말이다. 그러므로 형태론의 차원에서 분석하되 어떤 정도로 할 것인가가 결정되어야 할 문제이다.

한국말의 띄어쓰기는 이미 상당히 깊은 정도로 형태론적인 분석을 시행한 결과인 만큼 그대로 받아들이는 것이 여러 모로 편리하다. 그러니까 띄어쓰인 단위 곧 어절에서 얼마큼이나 더 들어갈 것인가가 실질적인 문제이다. 형태론적인 처리를 다양하게 하기 위해서는 형태 분석을 철저히 하는 것이 필요하다. 그러나 평면적인 분석에 치중하면 계층적인 처리에 어려움이 따르게 되므로, 그보다 높은 차원의 분석에도 도움을 줄 수 있도록 할 수 있는 한 많은 복수의 차원이 분석 방법에 반영되어야 한다.

그리고 언어 분석을 위한 말본 체계는 다소 무리한 경우가 따르더라도 일관성을 유지할 수 있도록 잘 선택하고 가다듬어 놓아야 한다. 실상 이 문제는 언어 연구의 전산화에 필수적인 전제 조건이다. 지금 어느 말본 체계도 완벽하지는 못하지만, 기계는 단순 논리에 지배되는 것이기 때문에

예외 없는 이론을 요구하는 것이다.

이러한 여러 가지 조건과 형편 및 편의를 고려해서 여기 예시할 만 한 언어 자료의 입력 방안은 다음과 같다. 분석된 단위에 매길 패는 이렇게 정해 본다 : ㅣ(이름씨), ㅐ(매인이름씨), ㅎ(홀로이름씨), ㄷ(대이름씨), ㅅ(셈씨) ; ㅓ(어찌씨), ㅁ(매김씨) ; ㄴ(느낌씨), ㅡ(이음씨), ㅜ(움직씨), ㄱ(그림씨), ㅈ(잡음씨), ㅍ(도움풀이씨) ; ㅌ(토씨). 이들과 함께 여러 가지 참조 사항을 나타낼 패도 필요하다 : ㄲ(권), ㅇ(장), ㄹ(절), ㅉ(쪽), ㅊ(참고) 등등. 이런 요령과 베이직 언어로 실제의 글을 입력하면 다음과 같이 된다.

```
10 DATA 요한=복ㅡ음., ㅎ, 1, ㅇ, 1, ㄹ, 태ㅡ초~에, ㅣㅌ, 말ㅡ씀~이, ㅣ
ㅌ, 계ㅡ시ㅡ니ㅡ라., ㅜ, 이, ㅁ, 말ㅡ씀~이, ㅣㅌ, 하나ㅡ님~과, ㅎㅌ, 함께,
ㅓ, 계ㅡ시ㅡ었ㅡ으니, ㅜ, 이, ㅁ, 말ㅡ씀~은, ㅣㅌ, 곧, ㅓ, 하나ㅡ님~이ㅡ샤
ㅡ니ㅡ라., ㅎㅈ, 2, ㄹ, 그~가, ㄷㅌ,
12 DATA 태ㅡ초~에, ㅣㅌ, 하나ㅡ님~과, ㅎㅌ, 함께, ㅓ, 계ㅡ시ㅡ었ㅡ고,
ㅜ, 3, ㄹ, 만ㅡ물~이, ㅣㅌ, 그~로, ㄷㅌ, 말미=암ㅡ아, ㅜ, 지ㅡ은, ㅜ, 바,
ㅐ, 되ㅡ었ㅡ으니, ㅜ, 지ㅡ은, ㅜ, 것~이, ㅐㅌ, 하나~도, ㅅㅌ, 그~가, ㄷ
ㅌ, 없ㅡ이~는, ㅓㅌ, 되ㅡㄴ, ㅜ, 것~이, ㅐㅌ, 없ㅡ느ㅡ니ㅡ라., ㄱ, ***, ***
```

이렇게 해 놓으면, 음소와 음절은 물론이오, 형태소의 단위로부터 어절 단위로, 또는 그 이상으로 분석의 차원을 높여 갈 수도 있을 것이다.

(2) 처리

입력된 자료의 처리는 주로 찾아내기와 가르기를 위한 것이다. 찾아내기[searching]란 특정한 자료를 지정해서 그것이 들어 있는 자리와 문맥 같은 것을 알아 내는 일이고, 가르기[sorting]란 일정한 부류의 자료를 찾아낸 다음에 일정한 기준에 따라 예를 들면 가나다순 같은 것으로 분류하고 배열하는 일이다.

위의 예문 가운데서 본보기로 매김씨 "이"를 찾아내는 프로그램과 그 결과는 다음과 같이 된다.

```
20 REPEAT
```

```
30 READ DT$, ID$
40 IF ID$ ="ㅇ" THEN CH=VAL(DT$)
50 IF ID$ ="ㄹ" THEN VS=VAL(DT$)
60 IF (DT$ ="이")*(ID$ ="ㅁ") THEN PRINT DT$ ; "(" ; CH ; " : " ;
VS ; ")" ; " " ;
70 UNTIL DT$ ="***"
```

화면의 결과 :

```
이(1 : 1) 이(1 : 1)
```

예문 가운데 어찌씨를 모두 찾아내게 하려면 60번을 다음과 같이 고치면 된다.

```
60 IF LEFT$ (ID$, 1) ="ㅓ" THEN PRINT DT$ ; "(" ; CH ; " : " ; VS ; ")"
   ; " " ;
```

화면의 결과 :

```
함께(1 : 1) 꼴 (1 : 1) 함께(1 : 2) 없—이～는(1 : 3)
```

물론 이것은 지극히 초보적인 사례에 지나지 않는다. 아주 복잡하고 다채로운 결과를 원하는 대로 얻을 수 있게 해 주는 것이 전산기인 만큼, 일단 착수하고 나면, 그 정확하고 충직한 심부름꾼은 달리 얻을 수 없을 것이다. 사람도 제 능력을 다 써 보지 못하고 가거니와, 전산기도 마찬가지이다.

4

끝으로, 개인용 전산기를 막상 선정하는 일은 여간 어렵지 않다. 워낙 다양한 기종이 나와 있는 데다가 끊임없이 놀라운 속도로 개량되고 값이 내려가기 때문이다. 일반적으로 말해서 힘이 닿는 한은 지금 가장 좋은 것, 특히 언어 연구를 위해서는 처리 속도가 빠르고 기억 용량이 큰 16비트 또는 32비트 전산기 가운데서 선택하는 것이 안전하다. 보조 기억 장치로는 적어도 40메가바이트 이상을 담을 수 있어야 할 것 같다. *