

## 자리채우미 (Placeholders)

박진호  
한양대학교

A placeholder is an element which fills in a position in a sentence instead of another prototypical element. It is used when the exact referring expression is not known, when the speaker wants to avoid explicit mention of it, or when the identity of the referent is not a question. Sometimes a placeholder is required by syntactic rules, or in a translation process, it is needed by the fact that the translation equivalent in the target language is a bound element. It also appears when a part of a discourse is omitted. Formal languages have placeholders, too.

### 1. 서론

하나의 완성된 문장을 만들기 위해서는, 우선 그 문장을 구성하는 단어들을 선택해야 한다. 단어들을 선택하고 나면, 해당 언어의 통사규칙과 각 단어의 어휘적 속성을 존중하면서 단어들을 결합하여 문장을 형성하게 된다. 그런데, 만들고자 하는 문장의 특정 위치에 들어갈 단어의 품사라든지 의미부류는 대체로 결정하였지만 정확히 어떤 단어인지까지는 결정하지 못한 경우가 있을 수 있다. 이럴 때, 이 한 단어 때문에 문장 전체를 발화하지 않을 수도 있겠지만, 어떤 경우에는 이 한 단어가 들어갈 자리를 임시방편으로라도 어떻게든 채워서 문장을 발화해야 할 경우도 있을 수 있다. 이 때 임시방편으로 채워지는 요소를 placeholder 라고 한다. 본고에서는 placeholder 를 ‘자리채우미’라고 번역한다. 본고는 자리채우미가 어떠한 경우에 사용되는지, 자리채우미의 특징이 무엇인지 등을 언어학적 관점에서 살펴보는 것을 목표로 한다.

### 2. 자리채우미의 기능

앞에서 이야기한 대로, 자리채우미는 문장 안에 들어갈 특정 단어가 생각나지 않을 때 사용될 수 있다.

- (1) a. 개 이름이 뭐더라 ... 아무튼 내가 어제 누굴 만났는데 ...  
b. 내가 서점에서 그 작가의 책을 봤는데 제목이 ‘무슨무슨 리포트’였어.

(1a)는 화자가 어제 만난 사람의 이름이 생각나지 않아서, 그 사람의 이름 대신에 ‘누구’라는 비한정 대명사(indefinite pronoun)를 사용한 것이다. (1b) 역시 책 제목 중 일부만 생각나고 일부는 생각나지 않아서, 생각나지 않는 부분을 ‘무슨무슨’이라는 비한정 관형사(indefinite adnominal)로 표현한 것이다.

문장 내의 특정 부분을 일부러 비워 두기 위해 자리채우미가 사용되기도 한다.

- (2) a. 정○○은 한국 모더니즘 시를 한층 발전시켰다.  
b. 김 의원과 박 장관은 어제 서울 시내 모 음식점에서 만난 것으로 확인되었다.

## 2 박진호

(2a)는 월북 작가들이 해금되기 전의 문학개론 교과서에서 월북 작가의 구체적인 이름을 밝히는 것을 피하기 위해 姓만 명기하고 이름 부분은 ‘○○’로 처리한 예이다. (2b)는 신문 기사에서 김 의원과 박 장관이 만난 사실을 보도할 때, 두 사람이 만난 정확한 장소를 밝히는 것을 피하면서도 만남의 구체적인 정황을 제시하여 독자들에게 신뢰감을 주기 위해, 음식점의 이름을 밝히는 대신 ‘모’라는 비한정사를 사용한 것이다.

문장 내의 특정 부분을 일부러 비워 두는 경우의 특수한 하위유형으로서 다음과 같은 것이 있다.

- (3) a. They depended ( ) him. / They depended \_\_\_\_ him.  
b. 독일에는 □ 하는 엘리베이터가 있다.

(3a)는 영어 시험 문제에서 괄호 안에 알맞은 단어를 집어넣는 이른바 괄호넣기 문제이다. 시험의 성질상 해당 단어를 수험생에게 알려주면 안 되기 때문에, 그 부분을 괄호나 밑줄로 비워두어야 한다. (3b)는 TV 프로그램에서 출연자들에게 퀴즈를 내 주고 그에 대한 해답을 추측하게 하는 경우에, 퀴즈의 핵심 부분을 □로 처리한 것이다. 진행자는 이것을 ‘네모’라고 읽는다.

위에서 살펴본 첫째 유형은, 지시대상의 identity는 알고 있지만 그것을 가리키는 정확한 지시표현(referring expression)을 모르는 경우이고, 둘째 유형은 지시대상의 identity도 알고 있고 그것을 가리키는 지시표현도 알고 있지만 일부러 정확한 지시표현을 회피한 경우였다. 다음으로 살펴볼 셋째 유형은, 지시대상의 정확한 identity가 무엇인가 하는 것이 문제가 되지 않는 경우이다.

- (4) a. ‘누가 무엇을 하다’ 같은 형식의 문장을 타동문이라고 한다.  
b. ‘누구누구에게’에서 ‘-에게’를 여격조사라고 한다.

(4a)는 타동문이 무엇인지를 설명하기 위한 문장으로서, 타동문이 주어와 목적어를 가진다는 것만 보여주면 그만이지, 주어와 목적어의 지시대상이 무엇인지는 전혀 문제가 되지 않는다. 그래서 주어 및 목적어 자리에는 비한정 대명사를 집어넣었다. (4b)는 여격조사를 설명하기 위한 문장으로서 초점은 ‘-에게’에게 맞춰져 있지만, ‘-에게’가 독립적으로 쓰일 수 없고 체언에 붙어야 하기 때문에 host 역할을 할 ‘누구누구’를 집어넣은 것이다.

셋째 유형의 특수한 경우로서, 관공서나 은행 등에서 문서 양식 견본을 진열할 때, 성명란에 ‘홍길동’, ‘김갑동’, ‘김철수’라고 쓴다든지, 전화번호란에 ‘123-4567’, ‘000-0000’이라고 쓰는 것 등을 들 수 있다. 문서를 어떻게 작성하는지를 예시한다는 목적상, 성명란에는 누가 보아도 사람 이름임을 알 수 있는 것을 써넣어야 하고, 전화번호란에는 전화번호처럼 보이는 것을 써넣어야 한다. 그러면서도, 이것이 특정인의 진짜 이름이라거나 진짜 전화번호가 아니라 예시용이라는 느낌을 주면 더 좋을 것이다. 그래서 우리 문화에서 누구나 알 수 있는 유명하거나 흔한 이름을 써 넣고, 비현실적인 전화번호를 써넣는 것이다. 이 유형은, 정상적인 문장을 구성하는 경우는 아니지만, 문서 양식이라는 특수한 텍스트를 구성하는 매우 중요한 요소라고 할 수 있으며, 그런 점에서 언어학적 논의의 대상이 될 수 있다.

넷째 유형은, 해당 언어의 통사 규칙 때문에 자리채우미가 사용되는 경우이다.

- (5) a. It rained yesterday.  
b. It is necessary for him to come here.

영어에는 주어 자리가 명사구 내지 명사구 상당 요소에 의해 채워져야 한다는 엄격한 통사 규칙이 있다. 그래서 (5a)에서 보듯이 rain 이 0 자리 술어이지만, 주어 자리에

허사(expletive)인 it 을 채워넣어야 한다. 그리고 (5b)에서 보듯이 주어 가 다른 위치로 이동하여 주어 자리가 비게 된 경우에도 역시 허사 it 을 집어넣어야 한다. 이런 예들은 허사라는 이름 아래 과거에 많이 논의되었지만, 자리채우미라는 관점에서 재조명될 필요도 있다고 생각된다.

넷째 유형에 들어갈 수 있는 것으로, 허사 주어 it 외에 동사도 있다.

- (6) a. Go on..
- b. The candle went out. / I put the candle out.
- c. I made off with money.
- d. I outdid him.

(6)은 Talmy(2000 vol.2: 284-286)에서 논의된 것인데, Talmy 의 주장의 핵심을 요약하면 다음과 같다. on, out, off, out- 같은 요소를 satellite 라고 하는데, 전세계 언어 중에는 사건의 핵심을 동사로 나타내는 verb-framed language 와 사건의 핵심을 satellite 로 나타내는 satellite-framed language 가 있다. 한국어는 전자의 예이고, 영어는 후자의 예이다. 영어에서 사건의 핵심이 satellite 로 표현되므로, 동사는 co-event(주사건의 방식, 원인 등에 해당하는 동반사건)를 나타내는 것이 일반적이다. 밖에서 안으로 들어가는 이동사건의 경우, 한국어는 ‘걸어서 들어가-’와 같이 주사건을 동사로 표현하고 동반사건을 부가어로 표현하는데, 영어는 ‘went in’과 같이 주사건을 satellite 로 표현하고 동반사건을 동사로 표현하는 것이다.

(6a)는 相(aspect)이라는 사건 유형의 예이다. ‘계속’이라는 주사건은 satellite 인 on 으로 표현되고 동반사건이 동사로 표현되는 것이 보통이다. 그래서 ‘이야기를 계속한다’는 의미를 나타낼 때 talk on 이라고 하는 것이 일반적이다. 그런데 화자는 동반사건이 무엇인지를 명시적으로 표현하지 않으려고 할 수도 있다. 그러나 그런 경우에도 영어의 통사 규칙상 온전한 문장이 되기 위해서는 동사가 필요하다. 이 때 go 라는 자리채우미가 동사 자리를 채워 주는 것이다.

(6b)는 상태변화라는 사건 유형의 예인데, 촛불이 꺼지는 상태변화는 satellite 인 out 으로 표현된다. 만약 바람이 불어서 촛불이 꺼졌다면 The candle blew out 이라 하면 되고, 내가 바람을 불어서 촛불을 껐다면 I blew the candle out 이라고 하면 된다. 그런데 촛불이 꺼진 원인, 내가 촛불을 끈 방법을 명시적으로 드러내지 않으려면 어떻게 해야 할까? 그러려면 blow 라는 동사를 빼야 하는데, 동사 없이는 온전한 문장이 안 되므로 그 자리에 자동문의 경우는 go, 타동문의 경우는 put 을 쓰는 것이다.

(6c)에서 satellite 인 off 는 ‘도망’을 의미하는데, 도망의 구체적인 방식이 동사로 표현되기도 한다. run off with money 는 돈을 가지고 달려서 도망가는 행위인 것이다. 그런데 도망의 구체적인 방식을 표현하지 않을 때에는 run 같은 구체적인 동사의 사용을 회피하게 되는데, 그렇다고 동사를 전혀 안 쓸 수는 없으므로, 그런 경우 자리채우미 동사인 make 를 집어넣는 것이다.

(6d)는 행위 상관관계(action correlation)라는 사건 유형의 예인데, 여기서 접두사 out-이 ‘능가’라는 주사건을 나타내며, 동반사건이 접두사 out-의 어기 동사로 표현된다. 그래서 I outplayed him 이라고 하면 ‘내가 그보다 연주를 더 잘 했다’라는 뜻이 된다. 동반사건을 구체적으로 표현하지 않으려 할지라도, 접두사가 host 없이 쓰일 수는 없으므로 자리채우미 동사 do 를 집어넣는 것이다.

(6)과 비슷한 유형으로서, 기능동사(support verb)라고 불리는 것들도 논의될 수 있을 것이다.

- (7) a. 철수는 하루종일 공부를 했다.
- b. 그가 드디어 결정을 내렸다.

#### 4 박진호

절의 의미상 술어(predicate)에 해당하는 요소가 동사로 실현되는 것이 가장 일반적이지만, 명사로 실현될 수도 있다. 그런데 이런 술어명사(predicate nominal)에는 시제·상·서법 요소가 붙을 수 없다. 따라서 술어명사를 통사적으로 support 해 주기 위해 허수아비 동사가 필요한 것이다. (7a)에서 술어명사 ‘공부’를 support 해 주는 ‘하-’, (7b)에서 술어명사 ‘결정’을 support 해 주는 ‘내리-’가 그런 요소이다. 이것을 대개 기능 동사라고 부른다. 본고의 관점에서는 이들을 자리채우미라고 할 수 있을 것이다.

다섯째 유형은, 번역 내지 번역 차용에서 자리채우미가 동원되는 경우이다. 번역을 할 때 source language 의 해당 단어와 의미상 가장 가까운 단어를 target language 에서 찾게 된다. 그런데 이 두 단어가 같은 품사일 수도 있지만 그렇지 않을 수도 있다. 특히 source language 의 자립적 요소의 번역 대당어(translation equivalent)가 target language 에서 의존적 요소일 수도 있다. 이 때 의존적 요소를 support 해 주기 위해 별 의미 없는 자리채우미가 동원될 수 있다.

구체적인 예로 한국어의 접속사 ‘그러나’, ‘하지만’, ‘허나’, ‘그러므로’, ‘그래서’, ‘그러면’, ‘허면’ 등을 들 수 있다. 한국어는 연결어미가 매우 발달해 있어서, 절과 절을 연결할 때 풍부한 연결어미를 이용하여 절과 절 사이의 다양한 의미관계를 나타낼 수 있다. 그래서 그런지 전근대 시기의 한국어 문헌에는 소위 접속사라는 것이 별로 나타나지 않고, 연결어미에 의해 절과 절이 연결되어 문장이 길게 늘어지는 것이 일반적이었다. 그러나 근대적인 문체가 정착되면서 문장의 길이가 짧아지게 되었고, 이에 따라 문장과 문장 사이의 의미관계를 표현할 장치, 즉 접속사에 대한 필요성이 증대하게 되었다. 여기에는 영어 등의 서양 언어들이 큰 영향을 미쳤을 가능성이 있다. 그래서 영어의 and, but, therefore, then 같은 접속사 내지 접속부사에 해당하는 단어를 만들어내게 된 것이다. 영어의 접속사 but 과 의미상 가장 가까운 한국어 요소는 연결어미인 ‘-나’, ‘-지만’이다. 그러나 이들은 의존요소이므로 자립적으로 쓰일 수 없다. 따라서 이들을 support 해 줄 host로서 별다른 의미가 없는 용언 ‘그렇-/그러-’나 ‘하/허-’를 짚어넣게 된 것이다.

한문을 번역한 중세 한국어 문헌에서도 이와 비슷한 예를 볼 수 있다. 『금강경삼가해』에서는 白話的 요소인 접속사 ‘爭奈’의 번역어로 ‘권마른’이 등장한다.

- (8) a. 비록 널오되 南北東西에 오직 이 내라 허나 권마른 一切 고대 어더 잡디  
 못허리로다 <金三 1:19b>  
 b. 悲로 舍生을 教化허샤미 곧 업디 아니허시나 권마른 能과 所왜 반득허거니와  
 智慧 | 眞際에 마자샤 平等허야 노프며 늦가오미 업스리라 <金三 2:13a>

이것은 용언 ‘-’에 역접의 연결어미 ‘-권마른’이 결합된 것이다. 현대 한국어의 ‘하지만’이나 ‘그러나’와 매우 비슷한 구조를 지니고 있는 것이다. 어느 경우든 용언은 자리채우미로서 사용되었다고 할 수 있다.

중세 한국어 문헌인 『두시언해』에서도 흥미로운 예를 발견할 수 있다.

- (9) a. 巴州入 사르미 常네 저기 골외면 蜀入 使者 | 된다마다 도라가디 못허놋다  
 <杜詩 11:50a>  
 b. 늘근 느치 된다마다 도트랏 平床을 어더 앉고 날회여 거로매 지즈로 대 막대를  
 어더 잡노라 <杜詩 15:15b>

한문에서 ‘動’은 동사로서는 ‘움직이다’의 의미이지만, 부사로 쓰일 때에는 ‘번번이, 걸핏하면, 그럴 때마다’를 의미한다. 이 부사적 의미에 가장 가까운 한국어 요소는 ‘-는다마다’라는 어미구조체이다. 그래서 부사 ‘動’의 번역어로 ‘-는다마다’를 사용하려 할 때, 이것이 의존요소이기 때문에 host 가 필요하게 된다. 『두시언해』에서 그 host 로

선택한 것은 동사 ‘뫼-’이다. ‘ㅎ-’ 같은 포괄동사(generic verb)를 쓰지 않고 ‘뫼-’를 선택한 것이 특이하다면 특이하다. ‘動-’이 동사로 쓰이는 것이 일반적이므로, 그에 해당하는 의미를 갖는 동사 ‘뫼-’를 선택한 것으로 보인다. 그러나 ‘뫼다마다’에서 동사 ‘뫼-’는 자리채우미로 쓰였을 뿐 ‘움직이다’의 의미를 나타내는 것은 아니다.

석독구결 자료의 ‘乃’자에서도 이와 비슷한 현상을 볼 수 있다.

- (10) a. 唯ハ 佛ス 與セ 佛スリニシ 乃シ {斯}リ 事乙 知ニテセ | (오직 부처와 부처라야 이 일을 아실 수 있다) <구인 11:24>
- b. 是 如 支 ヲ ハシ 乃シ 他ヲ 信施乙 受ノ 應 {應}セ ヲ 카 (이와 같아야만 남의 信施를 받을 만하다) <유가 17:20>
- c. {是}리 故 衆生乙 饒益 ヲ {爲}스 其 有セ 所乙 隨 一切之 皆セ 捨 ノ 乃 乃 ヲ 盡命 ヲ 至リ ヲ (이런 까닭으로 중생을 이롭게 하고자 가진 바에 따라 모두 버리되 내지 목숨을 버리는 것에 이르기까지 하나) <화소 10:11-13>
- d. 乃는 ㄷ ㅎ는 겨치라 (乃是 ‘-이야’ 하는 뜻의 토씨이다) <月釋 1:月釋序 13a>

(10d)는 한글 자료의 협주문에서 ‘乃’자의 뜻을 풀이한 것으로서, 한문의 ‘乃’에 가장 가까운 의미를 갖는 중세 한국어 요소가 ‘ㄷ’였음을 알 수 있다. 그런데 현대 한국어의 ‘-(이)야’가 완전한 의존요소인 데에 비해, 중세 한국어의 ‘ㄷ’는 부분적으로나마 약간의 자립성을 지녔던 듯하다. 그렇기에 (10d)와 같은 문장이 가능했을 것이다. (10a, b)에서 보듯이 석독구결 자료에서도 ‘乃’자에 ‘シ’가 현토된 것을 볼 수 있는데, (10a)에서 ‘シ’가 ‘乃’자의 안쪽에 표기되어 있는 것으로 볼 때 이것은 舍訓讀 표기로 추측된다. ‘乃’자의 온전한 訓을 ‘シ’로 표기한 것이다. 그런데 (10c)에서 보듯이 『華嚴經疏』의 현토자는 ‘シ’를 온전한 자립요소로 간주하지 않은 듯하다. ‘乃’자에 의미상 가장 가까운 요소가 ‘シ’이기는 하지만 이것이 의존적이기 때문에 host 가 필요하다고 판단한 듯하고, 그 host로서 포괄동사인 ‘-’를 선택하였다.

여섯째 유형은, 발화의 일부분을 생략할 때 생략된 부분 대신에 자리채우미를 사용하는 경우이다.

- (11) a. 영희 왈, 자기가 오늘 철수를 만났는데, 둘이 같이 학교엘 갔고, 그 다음에 찻집에서 차를 마셨고, 영화를 봤고 어찌구저찌구 말이 되게 많더라구.
- b. The minute she saw me, she said, “Oh, Aunt Mae, I hope it is OK that I wore my wedding dress. I didn't have time to shop and blah, blah, blah.” I was so disgusted, I walked away.

주로 발화의 어떤 부분을 세세하게 명시적으로 다 말하면 지루할 거라고 생각되거나 그 부분이 별로 중요치 않다고 판단될 때, 해당 부분을 ‘어찌구저찌구’나 ‘blah blah blah’ 같은 자리채우미로 대신하게 된다. 문어의 경우 해당 부분을 ‘...’ 같은 문장부호로 처리하기도 한다.

### 3. 형식언어의 자리채우미

프로그래밍언어나 논리 언어 등의 형식언어에서도 자리채우미가 사용된다. 논리 언어에서 람다 추상화(lambda abstraction)를 할 때, 명제 내의 특정 논항 자리에 x, y 같은 변항(variable)을 집어넣는다.

- (12)  $\lambda x [ P(\text{john}, x) ]$

(12)에서 술어 P 는 두 자리 술어인데, 그 중 첫째 논항 자리에는 john 이라는 상항(constant)이 들어갔지만 둘째 논항 자리에는 변항 x 가 들어가 있고, 이 변항 x 를 가지고서 람다 추상화가 이루어졌다. 따라서 (12)의 논리 표현은 온전한 명제가 아니라 한 자리 술어의 역할을 하게 된다. 이 때 변항 x 는 두 자리 술어인 P 의 둘째 논항 자리를 채워주는 자리채우미로 기능하고 있는 것이다.

프로그래밍 언어에서도 이와 비슷하게 자리채우미가 사용된다. 예컨대 Boost C++ 라이브러리에 포함되어 있는 Lambda 라이브러리나 Bind 라이브러리에서는 어떤 함수의 argument 자리에 자리채우미를 집어넣어 그 함수의 evaluation 을 지연시킬 수 있게 하고 있다(<http://www.boost.org/doc/html/lambda.html>, <http://www.boost.org/libs/bind/bind.html>).

```
(13) a. for_each(a.begin(), a.end(), std::cout << _1 << ' ');
      b. bind(f, 5, _1)(x);
```

C++ 표준 라이브러리의 알고리즘인 for\_each 는 3 개의 argument 를 취하는데, 첫째 argument 부터 둘째 argument 까지 셋째 argument 를 적용하라는 의미이다. 이 때 셋째 argument 에는 1 자리 함수나 그에 상응하는 function object 가 와야 한다. (13a)의 std::cout << \_1 << ' ' 라는 표현에서 '\_1' 자리에 "hello" 같은 일반적인 문자열이 오면 이 표현은 0 자리가 되는데, 이 자리에 자리채우미인 '\_1'이 옴으로써 1 자리 function object 가 만들어진다. (13b)는 두 자리 함수인 f 의 첫째 argument 자리에 5 를 집어넣고 둘째 argument 자리에는 자리채우미 \_1 를 씌으로써 bind 한 것이다. 따라서 bind(f, 5, \_1)는 1 자리 function object 가 된다. 그래서 나중에 x 를 argument 로 취할 수 있는 것이다. (12)와 (13)의 용법은 통사적 규칙 때문에 자리채우미가 사용되었다는 점에서, 2 절에서 살펴본 넷째 유형과 비슷하다.

위의 경우와는 좀 다른 차원에서 자리채우미가 사용되기도 한다. 프로그래머가 코딩을 할 때 identifier 는 마음대로 정해서 쓸 수 있다. 대개는 자신 및 다른 프로그래머가 identifier 를 보고서 어떤 용도로 쓰이고 있는지를 추측하기 쉽게 정해서 쓴다. 그런데 프로그래밍을 교육하거나 뉴스그룹 등에서 프로그래밍의 특정 주제에 대해 토의를 할 때, identifier 가 필요하기는 하지만 그 용도나 의미가 전혀 문제가 되지 않을 때에 'foo'나 'bar' 같은 무의미한 identifier 를 사용하는 일이 많이 있다.

```
(14) a. int foo(int x, int y);
      b. struct foo { void bar(); };
```

이것은 앞에서 살펴본, 문서 양식의 견본에 '홍길동' 같은 관습적 이름을 사용하는 것과 비슷한 용법이라고 할 수 있다.

### 참고문헌

- 박진호 2007. 유형론적 관점에서 본 한국어 대명사 체계의 특징. 국어학 50, 서울: 국어학회, 115-147.  
 Haspelmath, M. 1997. *Indefinite Pronouns*. Oxford: Clarendon Press.  
 Talmy, L. 2000. *Toward a Cognitive Semantics*. (2 vols) Cambridge: The MIT Press.

박진호  
 한양대 국어국문학과  
 성동구 행당동 17  
[sympjh@hanyang.ac.kr](mailto:sympjh@hanyang.ac.kr)