An HPSG Analysis of Persian Relative Clauses

Mehran Taghvaipour
University of Essex
matagh@essex.ac.uk

Relative clauses (RCs) in Persian are head-modifying constituents, all typically introduced by the invariant complementizer *ke*. Persian RCs are Unbounded Dependency Constructions (UDCs), containing either a gap or a resumptive pronoun (RP). The gap or the RP is linked to and licensed by the NP modified by the RC. In some positions only gaps are allowed, in other positions only RPs are allowed. There are also some positions where both gaps and RPs are alternatively allowed. Illustrating the striking similarities between Persian gaps and RPs, I will provide an HPSG unified approach to take care of the dependency between the licensing structure and the gap or RP with a truly single mechanism, using only the SLASH feature (as opposed to Vaillette (2001), who uses two different NONLOCAL features). Similar to Pollard and Sag's (1994) approach to the bottom of the dependency, I will assume a special *sign* at the bottom. However, my sign has a nonempty value for the SLASH feature and can be either a RP or a trace. I will introduce a feature called GAPTYPE which is a non-local feature and whose value can be either *trace* or *rp*. I will introduce two constraints to capture the pattern of distribution of RPs and traces. In the middle of the dependency, a trace or a RP is treated entirely in terms of the inheritance of the SLASH feature as in Sag (1997). At the top of the dependency, I will stop the non-empty SLASH at the complementizer point. I will propose a lexical entry for the complementizer *ke* that will account for the binding of SLASH by the feature BIND, which has a non-empty set as value.

1 The Data

Example (1) shows a restrictive RC in brackets.

(1)

zæn-i[1]	[*ke*	*mæn*	*dust+darœm*]
woman-RES	COMP	I	like-PRES-1sg

'The woman that I like'

Persian allows personal pronouns to behave resumptively. Example (2b) represents a Persian RC in which the pronoun *u*, i.e. 's/he', is used resumptively.

(2a)

mœrd-i	*ke*	*šoma*	____	*diruz*	*molaqat kœrdid …*
man-RES	COMP	you	Ø	yesterday	meet-PAST-2pl …

'The man whom you met yesterday…'

(2b)

mœrd-i	*ke*	*šoma*	***u***	*ra*[2]	*diruz*	*molaqat kœrdid …*
man-RES	COMP	you	**he**	RA	yesterday	meet-PAST-2pl …

'The man whom you met (***him**) yesterday…'

[1] This particle precedes restrictive RCs in Persian and is shown, henceforth, by RES in gloss.
[2] This particle (whose colloquial form is *ro*) is a specificity marker in Persian and is shown, henceforth, by RA in gloss.

The distribution of RPs and gaps in Persian RCs is shown in Table 1.

	Subject		Object of Prep.		Genitive		Direct Object	
	Restrictive	Non-Restrictive	Restrictive	Non-Restrictive	Restrictive	Non-Restrictive	Restrictive	Non-Restrictive
Gap is allowed?	Yes	Yes	No	No	No	No	Yes	No
RP is Allowed?	No	No	Yes	Yes	Yes	Yes	Yes	Yes

Table 1: Distribution of Gaps or Resumptive Pronouns in Persian RCs

Persian Gaps and RPs show striking similarities. I will provide a variety of evidence in favour of this similarity to conclude that Persian RCs contain traces, rather than null constituent gaps.

A strong argument in support of the fundamental similarity of RPs and gaps are comes from coordinate structures. Example (3) shows that in Persian a RP can be used with a gap in coordinate structures in unbounded dependencies. In fact, it is possible to have gaps in both conjuncts, RPs in both, or a gap in one conjunct and a RP in the other (in any order).

(3)
mærd-i *ke* ____ *pirahæn-e* *zærd* *pušideh+bud* *væ*
man-RES COMP Ø shirt-EZ yellow wear-PRESPART-3sg and
shoma *diruz* *az* ***u*** *pul* *qærz+gereftid* *Ali* *bud.*
you yesterday from **him** money borrow-PAST-2pl Ali was
'The man who___ was wearing a yellow shirt and you borrowed money from (***him**) was Ali.'

The second argument that supports the similarity between Persian RPs and gaps comes from parasitic gaps. Persian data shows that RPs, like gaps, can license parasitic gaps. I will bring examples (4a) and (4b) to illustrate this possibility. In (4a) there are two gaps, the second of which is parasitic. (4b) shows a sentence in which the second gap is still parasitic but licensed by the RP *un*.

(4a)
in *ketab-i-ye* *ke* *Yasmin* *bedun* *in* *ke* ___ *bexuneh* ___ *xærid.*
this book-RES-is COMP Yasmin without this COMP Ø read-3sg Ø bought-3sg.
'This is the book that Yasmin bought ___ without reading ___'

(4b)
in *ketab-i-ye* *ke* *Yasmin* *bedun* *in* *ke* ***un*** *ro* *bexuneh* ___ *xærid .*
this book-RES-is COMP Yasmin without this COMP **it** RA read-3sg Ø bought-3sg.
'This is the book that Yasmin bought (*it) without reading ___'

Another piece of supporting evidence for the similarity of Persian gaps and RPs is the sensitivity of RPs, like gaps, to certain islands. This is unlike what we see in Hebrew, for instance (see Vaillette (2001)). As an example, Persian gaps are sensitive to Subject Condition as shown in (5).

(5a)
[in ede'a ke Ali Hæmid ra dideh] Yasmin ra narahat+kærd.
[this claim COMP Ali Hamid RA seen Yasmin RA annoyed
'The claim that Ali has seen Hamid annoyed Yasmin.'

(5b)
**mærd-i ra ke [in ede'a ke Ali___/u ra dideh] Yasmin ra narahat+kærd.*
man-RES RA COMP [this claim COMP Ali___/ him RA seen] Yasmin RA annoyed.
'The man that the claim that Ali has seen ___ /him annoyed Yasmin.'

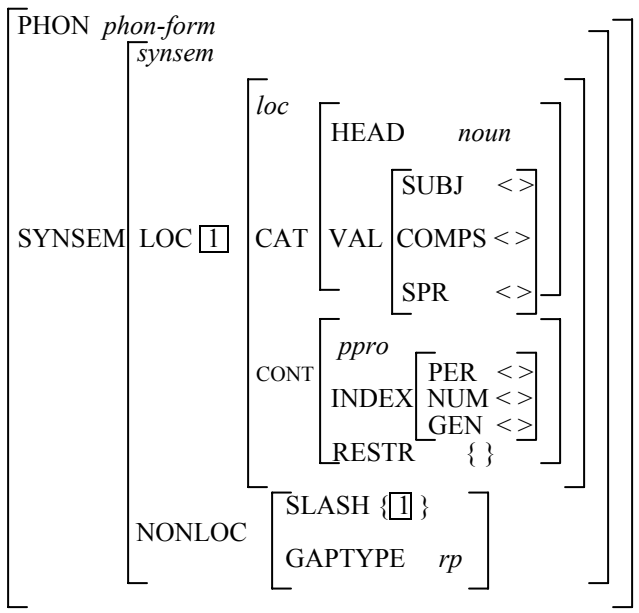
As I illustrated above, Persian gaps and RPs are strikingly similar: they have the same status within conjuncts, they can both license parasitic gaps; and, they are both sensitive to some island constraints. Based on this similarity, I will propose that they are both signs associated with the SLASH feature.

2 An HPSG Analysis

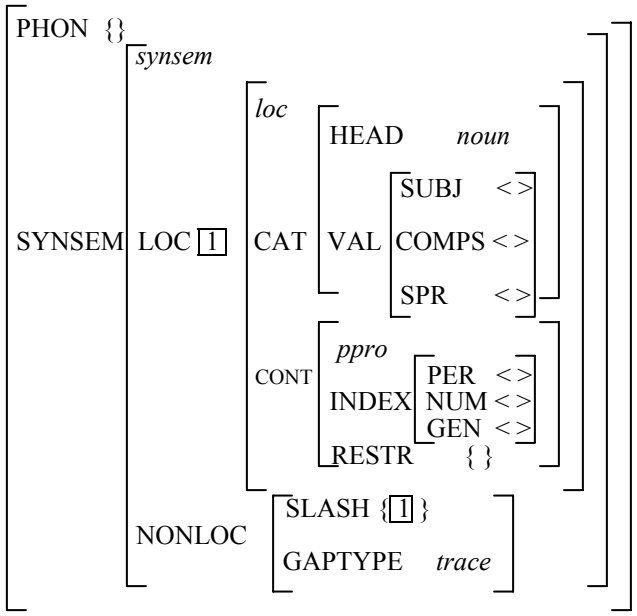
Bottom I will assume that the unbounded dependency in Persian RCs appear at the bottom of the dependency by a special sign that has a nonempty value for the SLASH feature. This special sign is either a trace or a RP. The nonempty SLASH feature encodes the information that there is a dependency between the trace/RP and the NP modified by the RC.

I will propose the lexical entry in (6) for RPs and the one in (7) for traces. The lexical entries in (6) and (7) are the same except in two respects. Firstly, the value of the PHON feature in traces is an empty set. This means that RPs as overt elements have phonology but traces do not. The second difference between these two lexical entries is that the value of their GAPTYPE features is different. GAPTYPE is a feature that I have introduced in order to capture the distributional properties of RPs and traces. GAPTYPE is a non-local feature whose value can be either *trace* or *rp*, for traces and RPs, respectively. The reason for distinguishing traces and RPs with a NONLOCAL feature is that this is not reflected within the value of SLASH and hence it is possible for a single unbounded dependency to be associated with a trace and an RP.

(6) Lexical Entry for a **resumptive pronoun**



(7) Lexical Entry for a **trace**



As for the pattern of distribution of RPs and traces, I will, first prevent RPs from appearing in subject position. I will propose the constraint in (8) to deal with this.

(8) [SUBJ <[1] >] → ~ ([1] = [SYNSEM|NONLOC|GAPTYPE *rp*])

The effect of this constraint is that if an element is in subject position, then the value of its GAPTYPE feature cannot be *rp*. In other words, if an element is a RP whose value of the GAPTYPE feature is *rp*, then it cannot come in subject position.

The second constraint, I will propose here, is to prevent traces from appearing in the positions of object of prepositions and possessors (i.e., in positions of the complements of non-verbs). This constraint is proposed in (9).

(9)

$$\begin{bmatrix} \text{HEAD} & [1] \\ \text{COMPS} & \langle \ldots, [\text{GAPTYPE } \textit{trace}], \ldots \rangle \end{bmatrix} \rightarrow [1] = \text{verb}$$

The effect of (9) is that if there is a trace as a complement of a head, then that head has to be a verb. Therefore, as in the case of object of preposition and genitive cases (possessors), the head is not a verb, we will not have a trace therein.

Middle In the middle of the dependency, I will follow Sag (1997). The SLASH is inherited by two constraints: Lexical Amalgamation of SLASH, and SLASH Inheritance Principle, given in (10) and (11) below.

(10) Lexical Amalgamation of SLASH

$$\textit{word} \Longrightarrow \begin{bmatrix} \text{BIND} & \boxed{0} \\ \text{ARG-ST} & \langle [\text{SLASH } \boxed{1}], \ldots, [\text{SLASH } \boxed{n}] \rangle \\ \text{SLASH} & (\boxed{1} + \ldots + \boxed{n}) - \boxed{0} \end{bmatrix}$$

(11) SLASH Inheritance Principle (SLIP):

$$\textit{hd-nexus-ph} \Longrightarrow \begin{bmatrix} \text{SLASH} / \boxed{1} \\ \text{HD-DTR} / [\text{SLASH } \boxed{1}] \end{bmatrix}$$

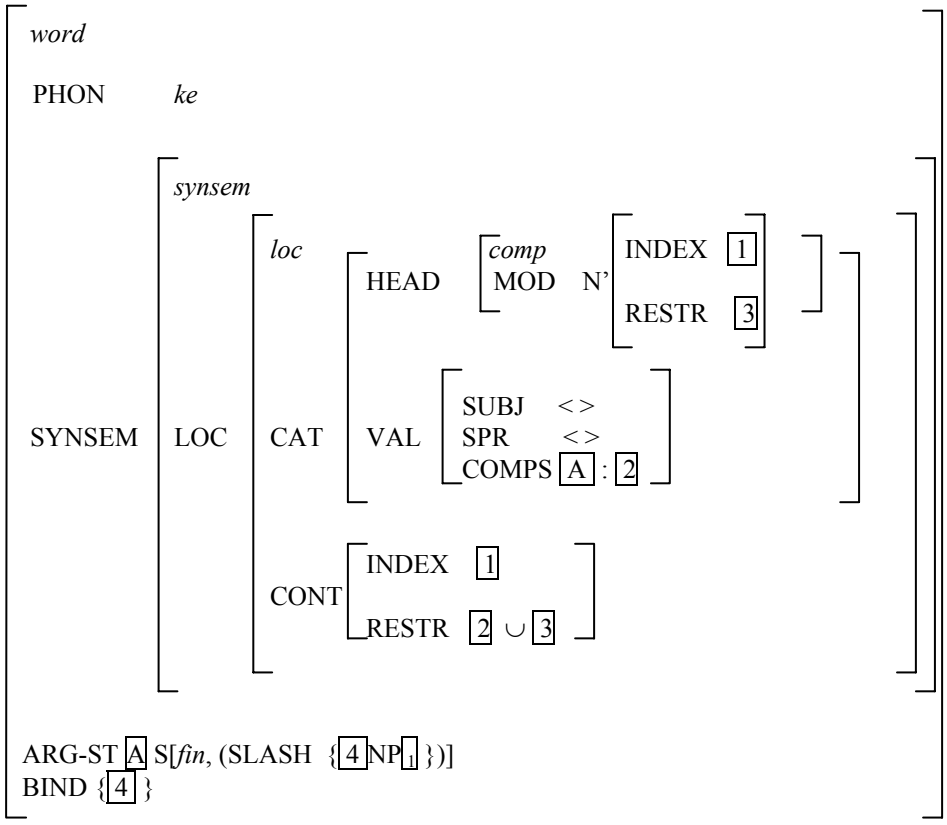
According to (10), all words, except SLASH binding elements like *tough*, specify empty value for the feature BIND. That is, in most cases nothing is subtracted from the disjoint union of the argument's SLASH values. Therefore, if a non-head-daughter is slashed so should the head daughter.

The constraint in (11) guarantees that the SLASH value of a phrase (of the type head-nexus-phrase) is the SLASH value of its head-daughter. In this way, any SLASH inheritance is mediated by the head-daughter, whose SLASH value contains that of the relevant non-head daughter.

Top At the top of the dependency, I will need some way to bind the SLASH feature. In other words, I will need a way to ensure that the non-empty SLASH value stops at an appropriate point.

This appropriate point, in Persian RCs, is the complementizer *ke*. I will propose the lexical entry in (12) for *ke* in RCs (i.e., ke_{RC}).

(12) Lexical Entry for ke_{RC}



The lexical entry for *ke* specifies some lexical information that ensures that the index of the N' (the NP modified by the RC) is identical to the SLASH value of *ke*. This structure-sharing, which is shown by tag 1, relates the trace or the RP to the NP modified by the RC. In addition, (12) also ensures that *ke* requires a sentential complement, shown by tag A. Tag A is the only member of *ke*'s ARG-ST list that stands for a finite sentence, containing a trace or a RP. The lexical binding of SLASH is accounted for by the feature BIND, which has a non-empty set as value for *ke*. This is shown by tag 4. The BIND feature will ensure that the trace or the RP is not amalgamated into the SLASH value of *ke* itself.

Selected References

- Pollard C., and I. Sag, 1994. Head-Driven Phrase Structure Grammar. The University of Chicago Press, USA.
- Sag, I., 1997. English Relative Clause Constructions. Journal of Linguistics 33:431-484.
- Vaillette, N. 2001. Hebrew Relative Clauses in HPSG. Proceedings of the 7th International Conference on Head-Driven Phrase Structure Grammar, CSLI Publications.