

PROPER NOUNS IN ST'ÁT'IMCETS (LILLOOET SALISH)

Henry Davis, University of British Columbia

1. Introduction. This paper investigates the syntax and semantics of proper nouns (PNs) in St'át'imcets (a.k.a. Lillooet Salish: ISO 639-3 lil). It argues for a dual representation of PNs. On the one hand, bare PNs resemble common nouns (CNs) in being lexically predicative: they take the full range of CN determiners and may be pluralized, modified, and act as the restrictions of quantifiers. On the other hand, when prefixed with the “nominalizer” *s-*, which occurs on PNs in both predicate and argument positions, PNs behave semantically as directly referring expressions: they take a special proprial determiner and cannot be pluralized, modified, or act as the restriction of a quantifier. It is argued that PNs are lexically represented either as predicates of type $\langle e, t \rangle$, in which case they function like CNs, or as arguments of type e , in which case they must be shifted into a predicative type in order to enter the composition, while maintaining their referential character: this is the function of the “nominalizer” *s-*, which under this analysis might actually be better termed a “predicativizer”.

2. Non-nominalized PNs. Without the nominalizer *s-*, PNs behave just like CNs. As predicates, they occur in the canonical clause-initial predicate position (following auxiliaries, if present), and cannot take determiners; their interpretation is that of a set of individuals.

- (1) *John*=ha=t'u7 ta=kúkwpí7=a láku7 tsal'álh=a?
John=Q=EXCL DET=John=EXIS there.INV Shalalth=EXIS
 'Is the chief over in Shalalth still a John?'¹

In argument positions, they take the full range of determiners (obligatory on CNs in argument position), can be modified, pluralized and quantified (2), and may occur as low-scope indefinites in the antecedent of conditionals and under negation (3).

- (2) wa7 láti7 gaw'-p=wít [tákem nelh=*Johns*=a] inátcwás
 IPFV there.VIS gather-INCH=3PL [all PL.ABSN.DET=*Johns*=EXIS] yesterday
 'All the Johns gathered there yesterday.'
- (3) cw7áoy=t'u7 káti7 [ku=*Johnny*]
 NEG=EXCL around.there.VIS [DET=*Johnny*]
 'There was no Johnny around there.'

3. Nominalized PNs. When introduced by the nominalizer *s-*, PNs behave quite differently. In predicate position, they refer to individuals, not to a set: compare (4) to (1):

- (4) *s-John*=ha=t'u7 ta=kúkwpí7=a láku7 tsal'álh=a?
 NMLZ-*John*=Q=EXCL DET=John=EXIS there.INV Shalalth=EXIS
 'Is the chief over in Shalalth still John?' (i.e., the individual John)

In core argument (subject and object) positions, nominalized PNs are optionally introduced by a special proprial determiner *kw=*:

- (5) ts7as cwíl'-en-Ø-as (kw=)*s-Laura*

¹ 'John' is a common last name as well as a first name amongst the St'át'imc.

come seek-DIR-3OBJ-3ERG (PN.DET=)NMLZ-Laura
 ‘He came looking for Laura.’

Nominalized PNs may not be modified or quantified, are ungrammatical in vocative and citation contexts, and can only be pluralized with the associative plural marker *wi* (otherwise used with numerals to mean ‘plus’):

(6) gaw’-p=wít [wi=s-Teresa]
 gather-INCH=3PL [ASSOC=NMLZ-Teresa]
 ‘Teresa and them (her relatives) met.’

4. Accounting for the differences. Non-nominalized PNs act straightforwardly as nominal predicates of type $\langle e, t \rangle$ (or its intensional counterpart), comparable in every way to CNs; this supports a predicative treatment of PNs. Nominalized PNs, in contrast, show the diagnostic properties of direct reference both in predicate and in argument positions. Note that this argues against reducing the referential status of PNs to the effects of the proprial determiner, as in Muñoz (2019), since the proprial determiner is present only in core argument positions, and even then only optionally, whereas nominalization itself is in strict complementary distribution with CN determiners, and is obligatory.

The most obvious way to treat this behavior is to treat all PNs as underlyingly predicative, and give the nominalizer the job of picking out a unique individual. Determiners will then do their usual job of converting the resulting predicates into arguments, with the proprial determiner restricted to predicates which denote unique individuals. Employing a predicativist lexical entry for a proper name such as ‘John’ as in (7), we might then give the nominalizer a lexical entry such as that in (8).

(7) $\llbracket \text{John} \rrbracket = \lambda x. x \text{ bears the name } /j\alpha n/$ (8) $\llbracket s- \rrbracket = \lambda P_{\langle e, t \rangle}. \lambda x_e : \exists! x [P(x) = 1] . P$
Only defined if P is a proper name

Within the grammar of St’át’imcets, however, nominalization also applies to a significant subset (about a third) of CNs. Clearly, when applied to CNs, the nominalizer cannot be picking out a unique individual, since CNs denote sets. For this reason, I follow an alternative line of analysis, motivated by the fact that St’át’imcets counts as a pure [+pred, -arg] system in the typology of Chierchia (1998): more specifically, *all nouns in St’át’imcets must enter the syntax as predicates*.

I propose that while CNs are lexically predicative (and thus the nominalizer is vacuous on them), PNs are lexically ambiguous between being predicative (as in (7)) and referential (as in (9)). In their referential guise, they must be converted to predicative status in order to enter the syntax, while retaining their referential character. I suggest that an operation essentially equivalent to Partee’s (1986) *Ident* is responsible for doing so, as in (10), yielding (11).

(9) $\llbracket \text{John} \rrbracket = \text{John}$ (10) $\llbracket s- \rrbracket = \lambda x_e \lambda y_e. [x = y]$ (11) $\llbracket s- \rrbracket (\llbracket \text{John} \rrbracket) = \lambda y_e [\text{John} = y]$
Only defined if x is a proper name

References: Chierchia, G. 1998. Reference to kinds across languages. *NLS* 6:339-405. Muñoz, P. 2019. The proprial article and the semantics of names. *S&P* 12(6):1-36. Partee, B. 1986. Noun phrase interpretation and type-shifting principles.