Contents

1 Function documentation 1
Bibliography 7
Index 9
Function documentation

cree_sro_syllabics\( \text{sro2syllabics} \) \( (\text{sro: str, hyphens: str} = \text{"u202f"}, \text{sandhi: bool} = \text{True}) \rightarrow \text{str} \)

Convert Cree words written in SRO text to syllabics.

Finds instances of SRO words in strings, and converts them all to syllabics.

```python
>>> sro2syllabics('Eddie nitisiyihkâson')
'Eddie ' 

You should be able to write words in Y-dialect (a.k.a., Plains Cree):

```python
>>> sro2syllabics('niya')
'\n'

...and Th-dialect (a.k.a., Woods Cree):

```python
>>> sro2syllabics('nitha')
'\n'

Any word that does not have the “structure” of a Cree word is not converted:

```python
>>> sro2syllabics('Maskêkosihk trail')
'Maskêkosihk ' 
>>> sro2syllabics('Maskêkosihk tireyl')
'Maskêkosihk ' 
```

Roman full-stops/periods ("."’) are converted into syllabics full-stops:

```python
>>> sro2syllabics('Eddie nitisiyihkâson.')
'Eddie ' 
```

Note that the substitution of full-stops only takes place after syllabics; if it is obviously not Cree (like most English), it will not be converted:
sro2syllabics() can handle variations in orthography. For example, it can convert circumflexes (âêîô):

```python
>>> sro2syllabics('êwêpîhkêwêpinamahk')
''
```

It can convert macrons (¯a¯e¯ı¯o):

```python
>>> sro2syllabics('¯ew¯ep¯ıhk¯ew¯epinamahk')
''
```

And it can convert an unaccented “e” just as if it had the appropriate accent:

```python
>>> sro2syllabics('ewepapihkewepinamahk')
''
```

Additionally, apostrophes are interpreted as short-i’s. For example, converting “tânsi” will not work as expected:

```python
>>> sro2syllabics("tânsi")
''
```

However, add an apostrophe after the ‘n’ and it will work correctly:

```python
>>> sro2syllabics("tân'si")
''
```

Hyphens in Cree words are replaced with `<U+202F NARROW NO-BREAK SPACE>` (NNBSP) by default. This is a space that is narrower than the normal space character. NNBSP also prevents breaking the word across line breaks. We chose the NNBSP character as the default, as it helps visually distinguish between meaningful sub-elements within words, while being less likely to be mistaken as word-separating whitespace by most text processing applications.

Compare the following hyphen replacement schemes:

<table>
<thead>
<tr>
<th>Replace hyphens with</th>
<th>kâ-mahihkani-pimohtêt isiyihkâsow</th>
</tr>
</thead>
<tbody>
<tr>
<td>(nothing)</td>
<td></td>
</tr>
<tr>
<td>NNBSP</td>
<td></td>
</tr>
<tr>
<td>Space</td>
<td></td>
</tr>
</tbody>
</table>

We discourage using an ordinary space character (U+0020), as it is often interpreted as separating words, both by computers and people alike. If you are viewing this documentation in a web browser, try double clicking the syllabics rendition of “kâ-mahihkani-pimohtêt” with NNBSP separators versus the one with space separators. Double clicking typically selects an entire word by default, and this is often the case when double clicking the word with NNBSP characters; however this fails for the rendition with space characters.

Despite this, you can chose any character of your liking to replace hyphens in syllabics by providing the `hyphens=` keyword argument:

```python
>>> sro2syllabics('kâ-mahihkani-pimohtêt', hyphens='\N{NARROW NO-BREAK SPACE}')
''
```

```python
>>> sro2syllabics('kâ-mahihkani-pimohtêt', hyphens='')
''
```

(continues on next page)
In SRO, the most orthographically correct way to write certain compounds is to separate two *morphemes* with a hyphen. For example:

- pîhc-âyihk — inside
- nîhc-âyihk — outside

However, both words are pronounced as if discarding the hyphen:

- pîhcâyihk — inside
- nîhcâyihk — outside

This is called *sandhi*. When transliterated into syllabics, the transcription should follow the latter, blended interpretation, rather than the former, separated interpretation. By default, `sro2syllabics()` applies the sandhi rule and joins the syllable as if there were no hyphen:

```python
>>> sro2syllabics('pîhc-âyihk')
''
```

However, if this is not desired, you can set `sandhi=False` as a keyword argument:

```python
>>> sro2syllabics('pîhc-âyihk', sandhi=False)
''
```

**Parameters**

- `sro` *(str)* – the text with Cree words written in SRO.
- `sandhi` *(bool)* – whether to apply sandhi orthography rule (default: True).

**Returns** the text with Cree words written in syllabics.

**Return type** str

```python
cree_sro_syllabics.syllabics2sro(syllabics: str, produce_macrons=False) → str
```

Convert Cree words written in syllabics to SRO.

Finds all instances of syllabics in the given string, and converts it to SRO. Anything that is not written in syllabics is simply ignored:

```python
>>> syllabics2sro('Eddie ')
'Eddie nitisiyihkåson.'
```

You should be able to convert words written in Y-dialect (a.k.a., Plains Cree):

```python
>>> syllabics2sro('')
'niya'
```

... and Th-dialect (a.k.a., Woods Cree):

```python
>>> syllabics2sro('')
'nitha'
```

By default, the SRO will be produced with circumflexes (âêîô):
This can be changed to macrons (âëî) by setting produce_macrons to True:

```python
>>> syllabics2sro('', produce_macrons=True)
'êwêpêhêwêpinamahk'
```

In both cases, the character produced will be a pre-composed character, rather than an ASCII character followed by a combining diacritical mark. That is, vowels are returned in NFC normalization form.

For compatibility with `cree_sro_syllabics.sro2syllabics()`, `syllabics2sro` will convert any instances of `<U+202F NARROW NO BREAK SPACE>` to a hyphen in the SRO transliteration.

```python
>>> syllabics2sro()  
'kâ-mahihkani-pimohtêt'
```

In some syllabics text, syllabics with a ‘w’ dot are rendered as two characters: the syllabic without the ‘w’ dot followed by `<U+1427 CANADIAN SYLLABICS FINAL MIDDLE DOT>`; this differs from the more appropriate pre-composed syllabic character with the ‘w’ dot. For example,

- pre-composed syllabic
- syllabic + CANADIAN SYLLABICS FINAL MIDDLE DOT

`syllabics2sro()` can convert both cases appropriately:

```python
>>> syllabics2sro()  
'itwêwina'
```

Some syllabics converters produce erroneous yet very similar looking characters. `syllabics2sro()` knows the following look-alike characters:

<table>
<thead>
<tr>
<th>Look-alike</th>
<th>Correct character</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANADIAN SYLLABICS WEST-CREE Y</td>
<td>CANADIAN SYLLABICS FINAL PLUS</td>
</tr>
<tr>
<td>CANADIAN SYLLABICS T</td>
<td>CANADIAN SYLLABICS WEST-CREE M</td>
</tr>
<tr>
<td>CANADIAN SYLLABICS SAYISI YI</td>
<td>CANADIAN SYLLABICS HK</td>
</tr>
</tbody>
</table>

Note: although `<U+1540 CANADIAN WEST-CREE Y>` is named as if it is the correct Y final, writers prefer `<U+1429 CANADIAN SYLLABICS FINAL PLUS>`. Both are in the shape of a plus sign, but FINAL PLUS is often rendered similarly to other finals, small, and above the center line, whereas WEST-CREE Y is rendered centered to other syllabics. Having all finals render small and above the centre line is preferable from most Cree writers I surveyed.

`syllabics2sro()` automatically interprets erroneous look-alikes as their visually equivalent characters.

```python
>>> syllabics2sro()  
'cîpêhtakwâpikwaniy'
>>> syllabics2sro()  
'asamopitan'
>>> syllabics2sro()  
'mâmihk'
```

**Parameters**

- `syllabics` *(str)* – the text with Cree words written in syllabics.
• **produce_macrons** – if True, produces macrons (äëîö) instead of circumflexes (âêîô).

**Returns** the text with Cree words written in SRO.

**Return type** str

### 1.1 Glossary

**Cans** The **ISO 15924** code for *Canadian Aboriginal syllabics*.

**circumflex** A **circumflex** is the little hat on top of a vowel (âêîô), sometimes used to represent long vowels in nêhiyawêwin.

**ISO 15924** The **ISO 15924** standard defines four letter codes for every writing systems in common use.

See also: ISO 15924 on Wikipedia, List of ISO 15924 codes.

**Latn** The **ISO 15924** code for the Latin writing system, including **Standard Roman Orthography**.

**macron** A **macron** is the little bar on top of a vowel (äëîö), sometimes used to represent long vowels in nêhiyawêwin.

**orthography** **Orthography** is the set of rules for writing a certain language. **Orthography** is Greek for “correct writing”.

**Plains Cree**

**Cree Y-dialect**

**nêhiyawêwin** Plains Cree, **Cree Y-Dialect**, or **nêhiyawêwin** is the most widely-spoken western Cree dialect. Notably, many words which have a “th” in **Woods Cree** and words that have a “n” in Swampy Cree have a “y” in Plains Cree; this is why Plains Cree is also known as the Cree Y-dialect.

**sandhi** Changes to pronunciation across word and morpheme boundaries. In Cree, this may occur when two morphemes are joined such as *pîhc- and -âyihk*. Written in SRO, it’s *pîhc-âyihk*. Sandhi in Cree means that the “c” in the first morpheme and the “ä” in the next morpheme should be joined and pronounced as one syllable. Hence, it is pronounced as *pîh-câ-âyihk*. The syllabics transliteration should be written as it is pronounced, thus *pîhc-âyihk* should be rendered as and not as -.

For a more thorough description of sandhi in Cree, see [Wolfart] and [Wolvengrey].

**SRO**

**Standard Roman Orthography** **Standard Roman Orthography** or **SRO** is a writing system for nêhiyawêwin that uses the Latin alphabet—the same alphabet as English and French. Initial attempts to write nêhiyawêwin in the Latin alphabet resulted in several different spellings, and several different ways to write the same thing [Okimäsis]. **SRO** is a unified way to write nêhiyawêwin Latin script. Even though **SRO** attempts to be consistent, there are still variations, such as the use of **circumflexes** (âêîô) or **macrons** (äëîö).

See also, **Beginning to read Plains Cree in Standard Roman Orthography**.

**syllabics**

**Canadian Aboriginal syllabics** **Syllabics** is a writing system for nêhiyawêwin (in syllabics, ) that uses symbols to represent full syllables. Cree syllabics form a part of the **Canadian Aboriginal syllabics** writing system family.

**transliterator**

**transcriptor**

**converter** A **transliterator** is a tool that converts between two different writing systems. Synonyms for “transliterator” include **converter** and **transcriptor**.

**Woods Cree**
Cree Th-dialect

nîhithawîwin  Woods Cree, Cree Th-Dialect, or nîhithawîwin is a Cree dialect spoken in Northern Saskatchewan and Northern Manitoba. It is distinct from Plains Cree in that many words that have a “th” in them are spoken with a “y” in Plains Cree; this is why Woods Cree is also known as Cree Th-dialect.
Bibliography


Index

C
Canadian Aboriginal syllabics, 5
Cans, 5
circumflex, 5
converter, 5
Cree Th-dialect, 6
Cree Y-dialect, 5
I
ISO 15924, 5
L
Latn, 5
M
macron, 5
N
nîhithawîwin, 6
nêhiyawêwin, 5
O
orthography, 5
P
Plains Cree, 5
S
sandhi, 5
SRO, 5
sro2syllabics() (in module cree_sro_syllabics), 1
Standard Roman Orthography, 5
syllabics, 5
syllabics2sro() (in module cree_sro_syllabics), 3
T
transcriptor, 5
transliterator, 5
W
Woods Cree, 5