A Shifting Perspective on Stem Space for French Verbs

Olivier Bonami & Gilles Boyé

Université Paris-La Sorbonne
LLF

Université de Bordeaux
CLLE

MorPa

Defining stem spaces

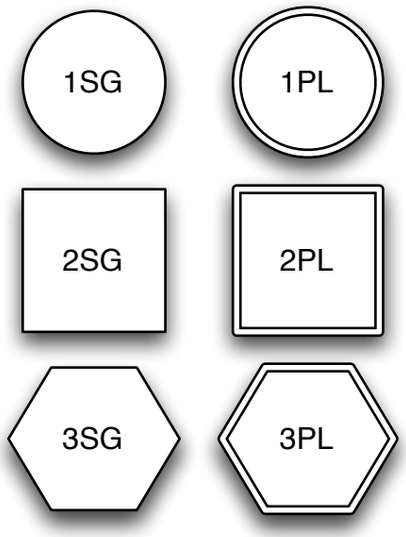
A Network of relations

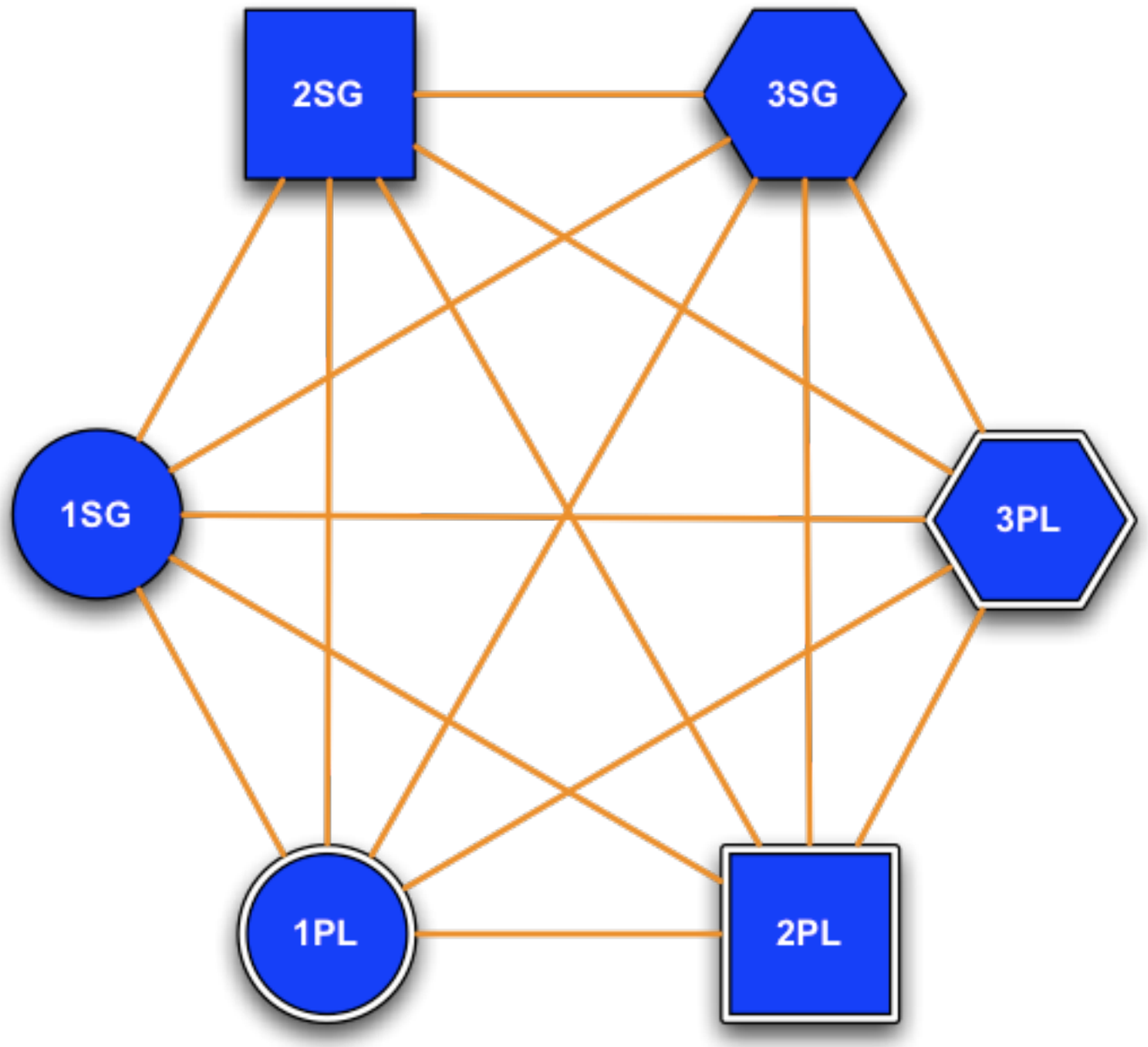
Motivations

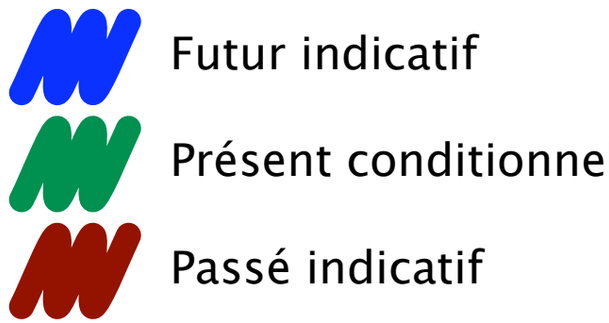
- In a Word and Paradigm framework, the lexeme is the unit binding together the various inflectional forms.
- In this context, two interesting conceptions:
  - From a psycholinguistic standpoint, can be seen as a network of relations linking every form to all the others in the same paradigm
  - From a linguistic standpoint, can be seen as a network linking all the forms of the paradigm to the principal parts.
- While the maximally connected network is easy to define, we show that it is not linguistically appropriate.
  - An appropriate network should constrain the distribution of irregularity to the observed patterns.
  - It should also accomodate different patterns of lexical storage for existing lexemes

Forms and paradigms

- A paradigm is a set of related forms

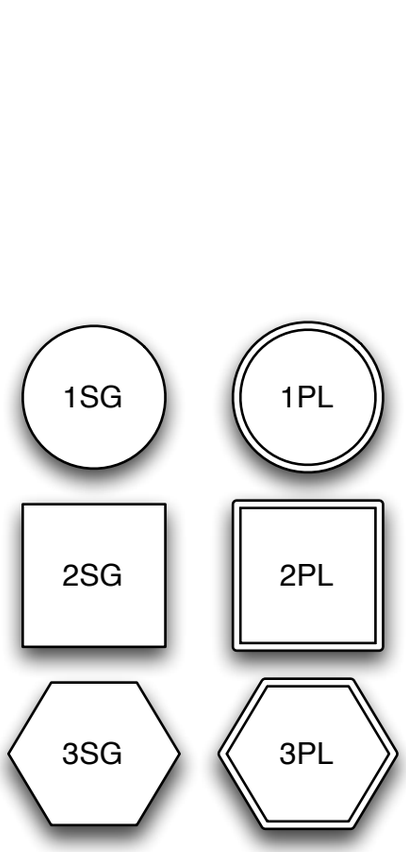


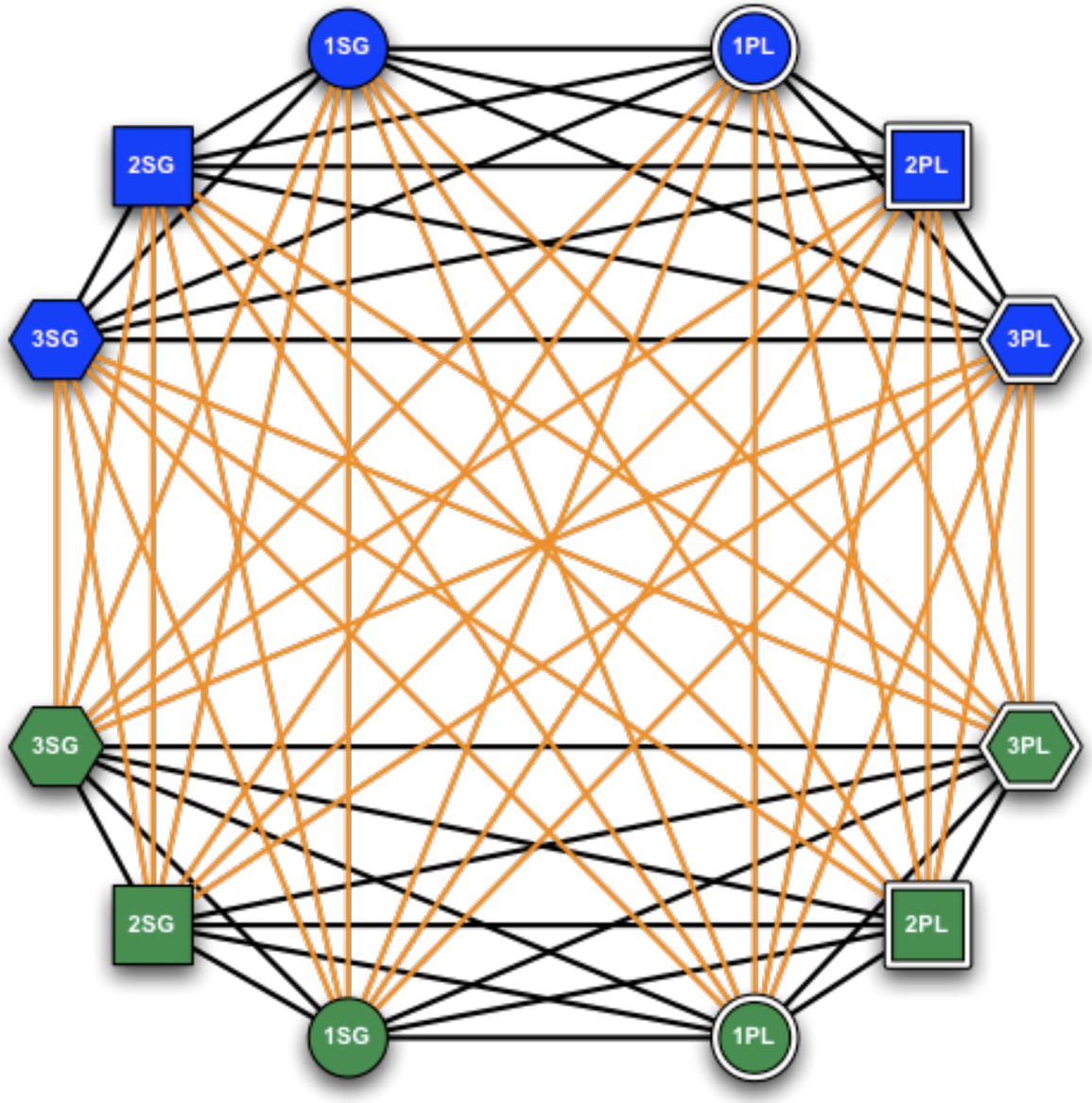




Forms and paradigms

- A paradigm is a set of related forms





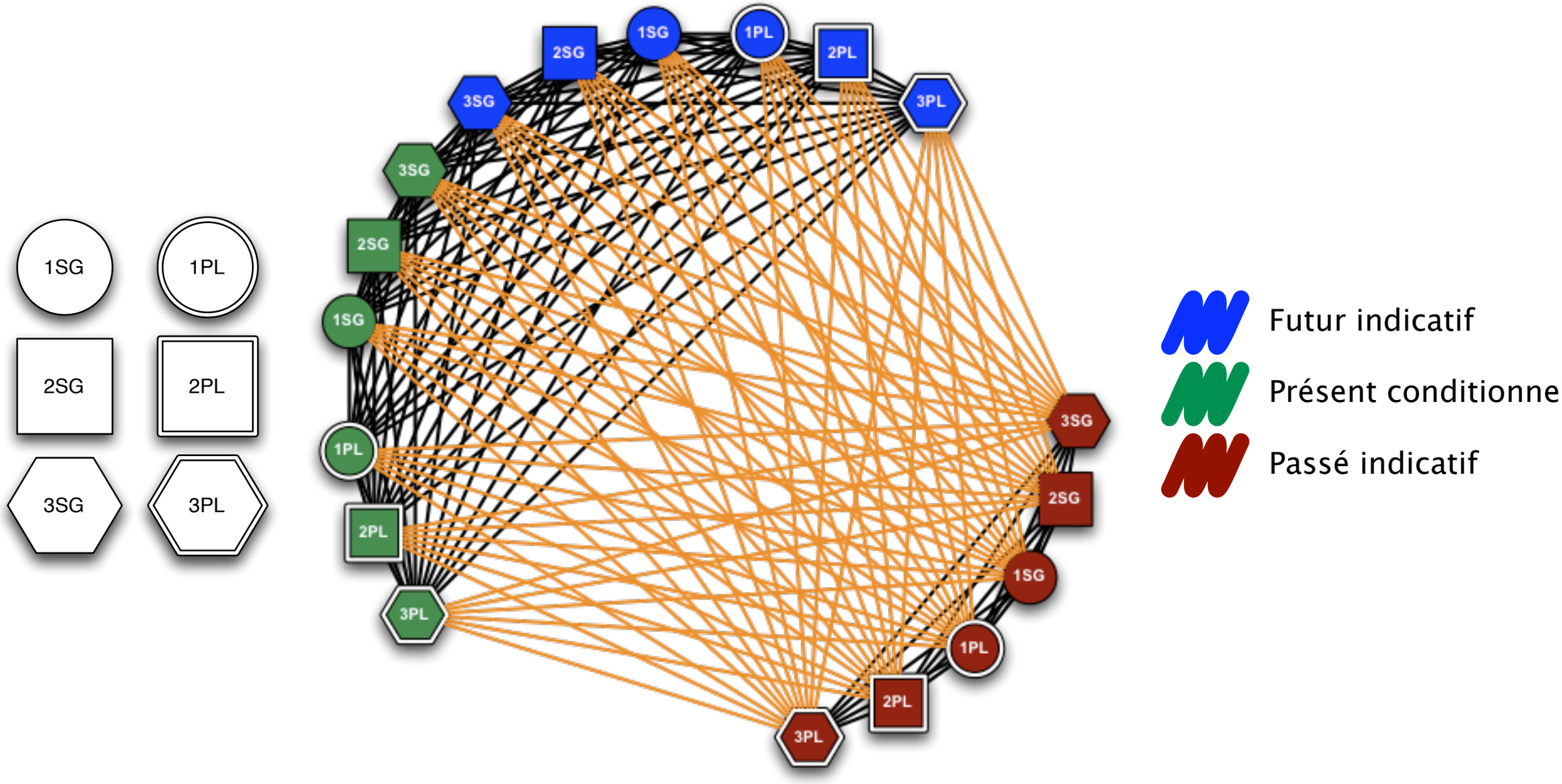
Futur indicatif

Présent conditionnel

Passé indicatif

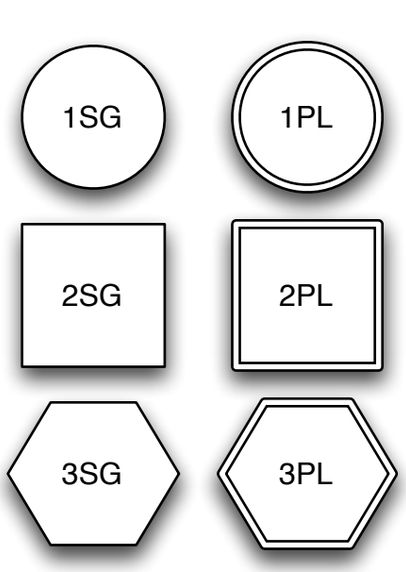
Forms and paradigms

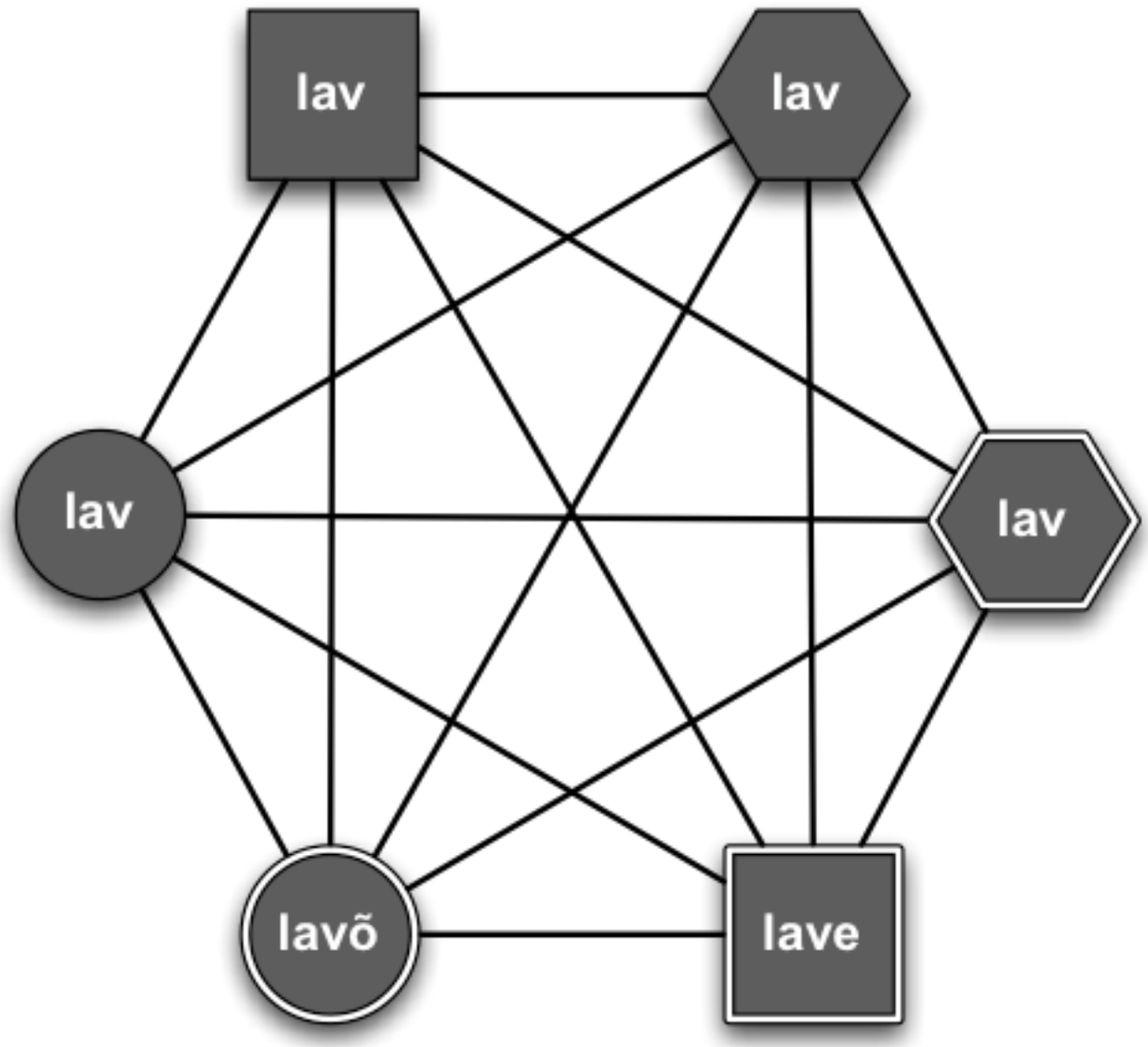
- A paradigm is a set of related forms



Points of irregularity

- To be an irregular verb is to necessitate more than one principal part
  - regular forms are connected



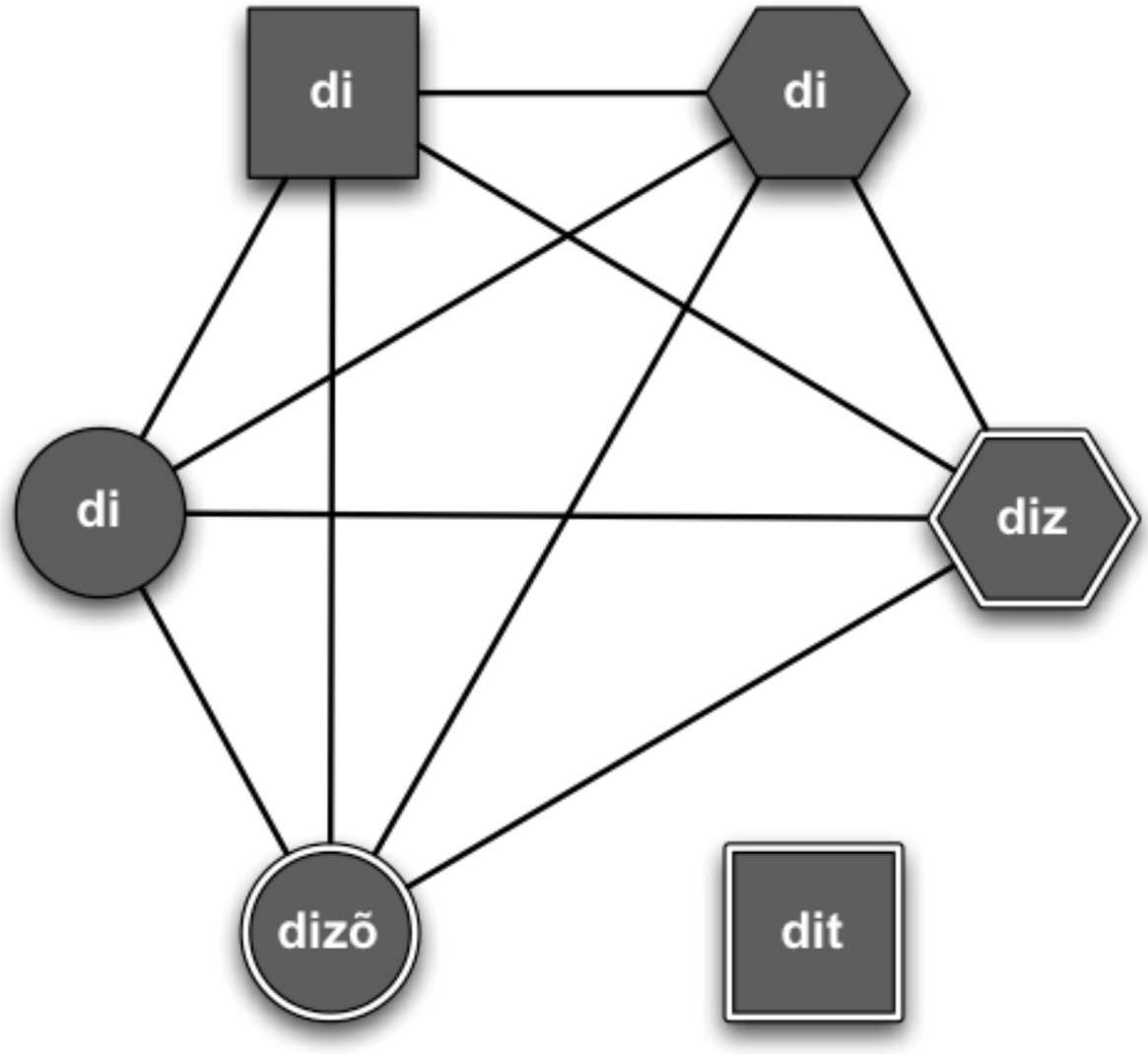


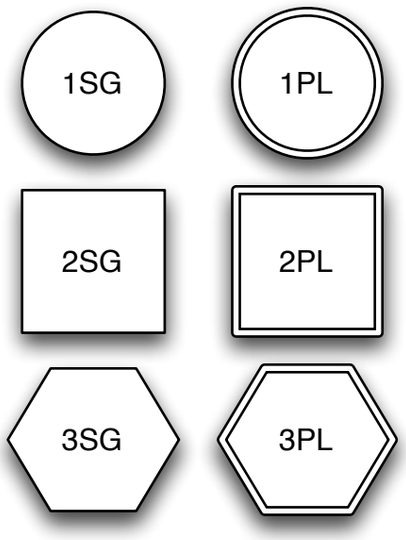
LAVER

☞ lavez

Points of irregularity

- To be an irregular verb is to necessitate more than one principal part
  - regular forms are connected
  - suppletive forms are isolated



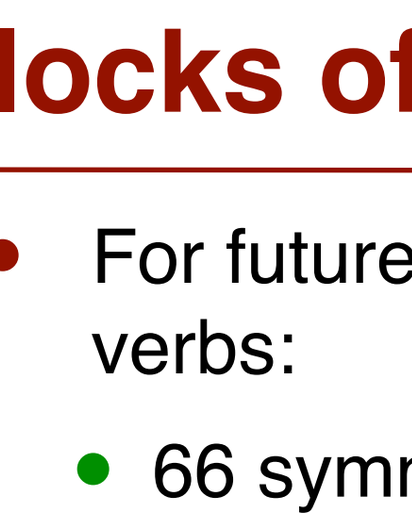


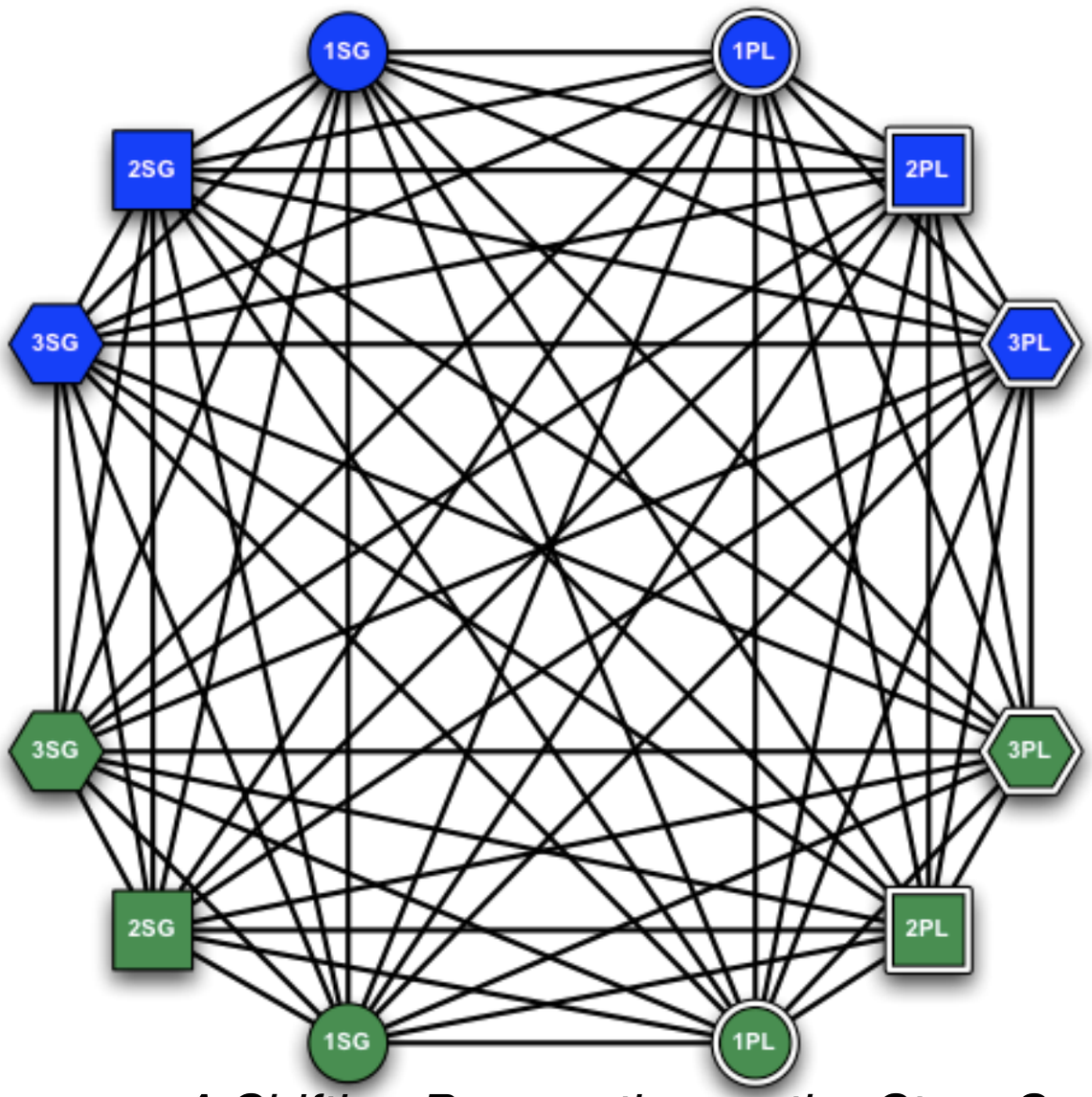
DIRE

☞ dites

Blocks of forms

- For future and conditional, the 12 forms are inter-predictable for all verbs:
  - 66 symmetric relations link all the forms together



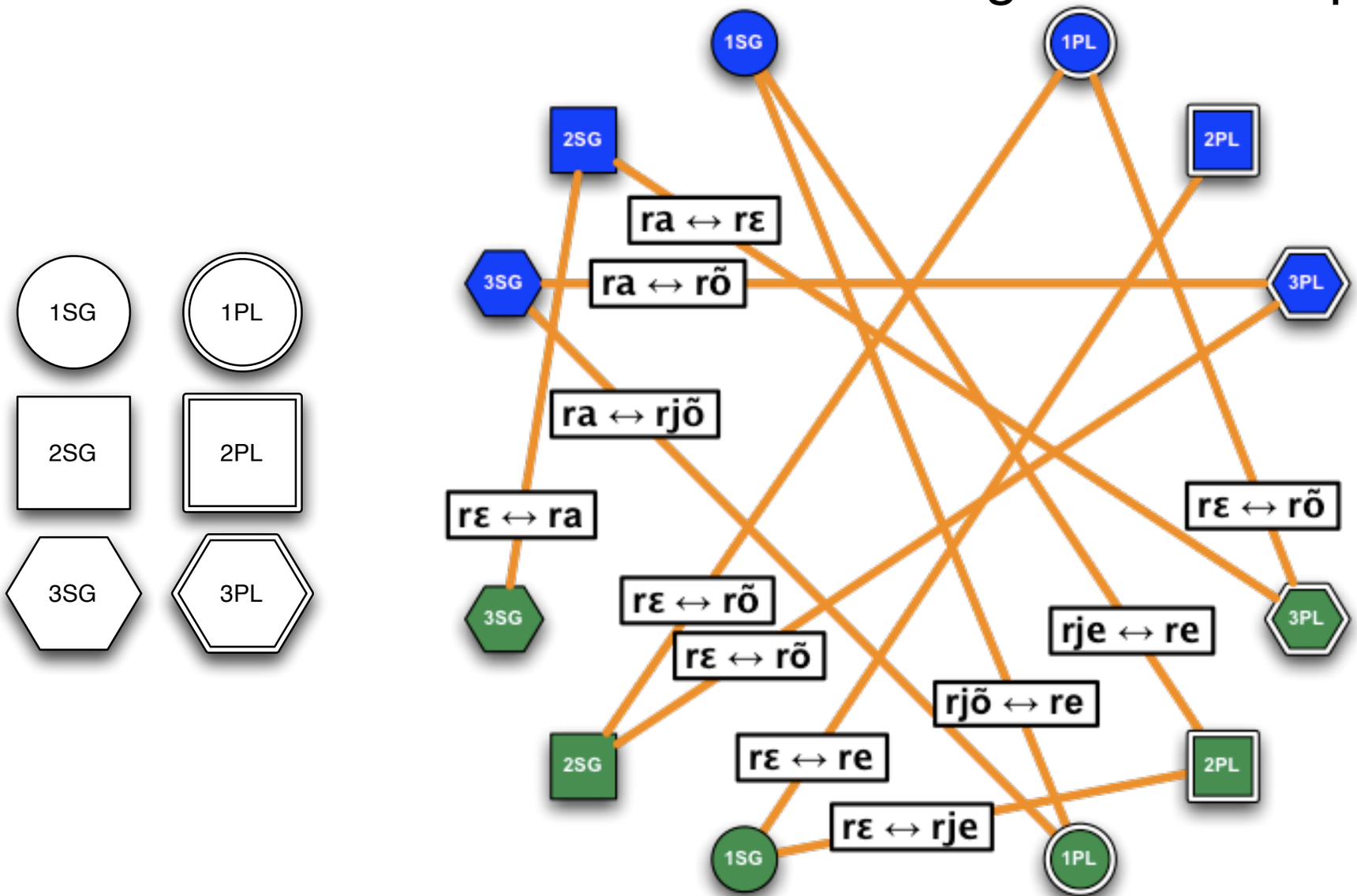


Futur indicatif

Présent conditionnel

Blocks of forms

- For future and conditional, the 12 forms are inter-predictable for all verbs:
  - 66 symmetric relations link all the forms together
  - 11 relations are sufficient for a linguistic description

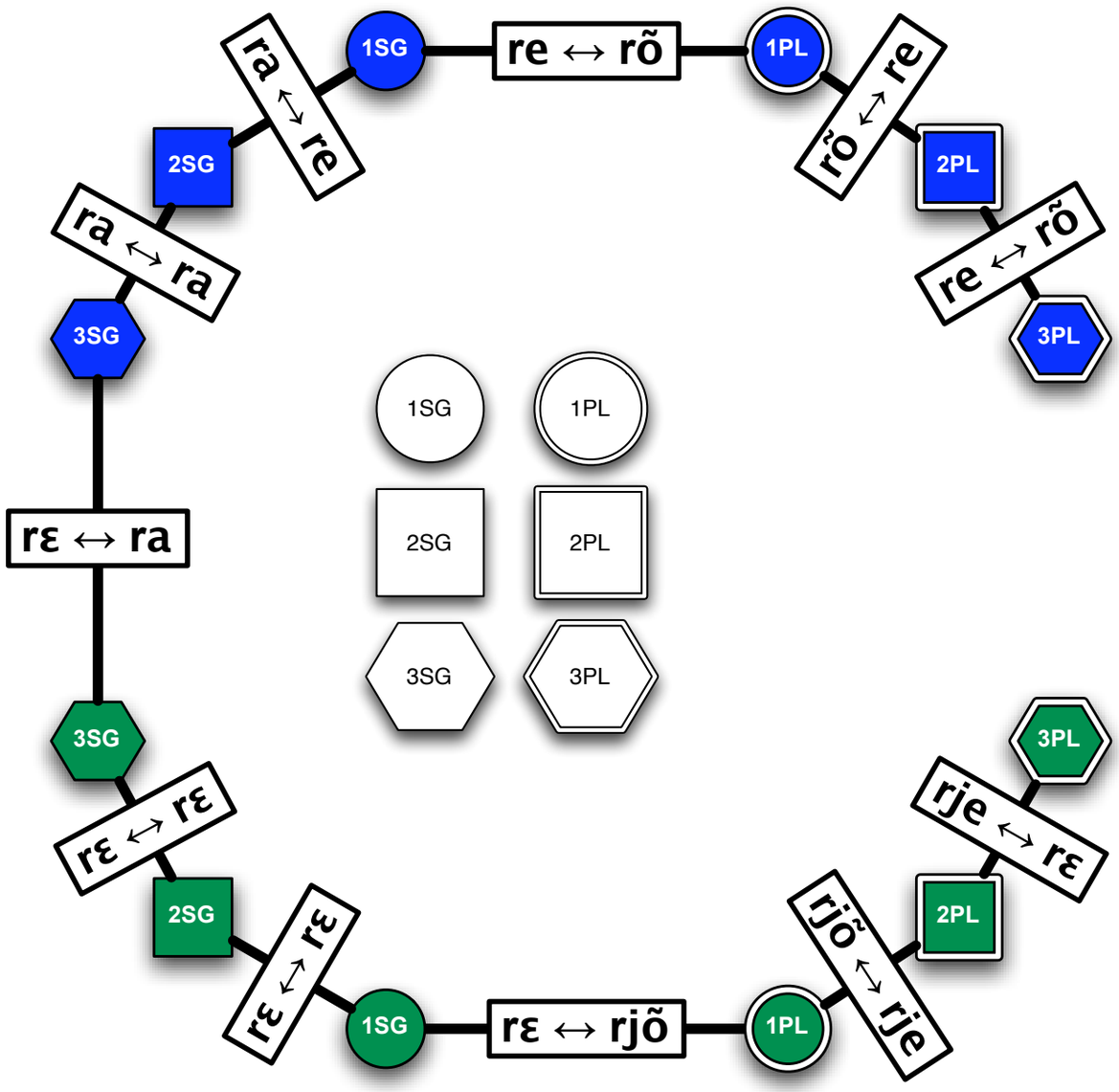


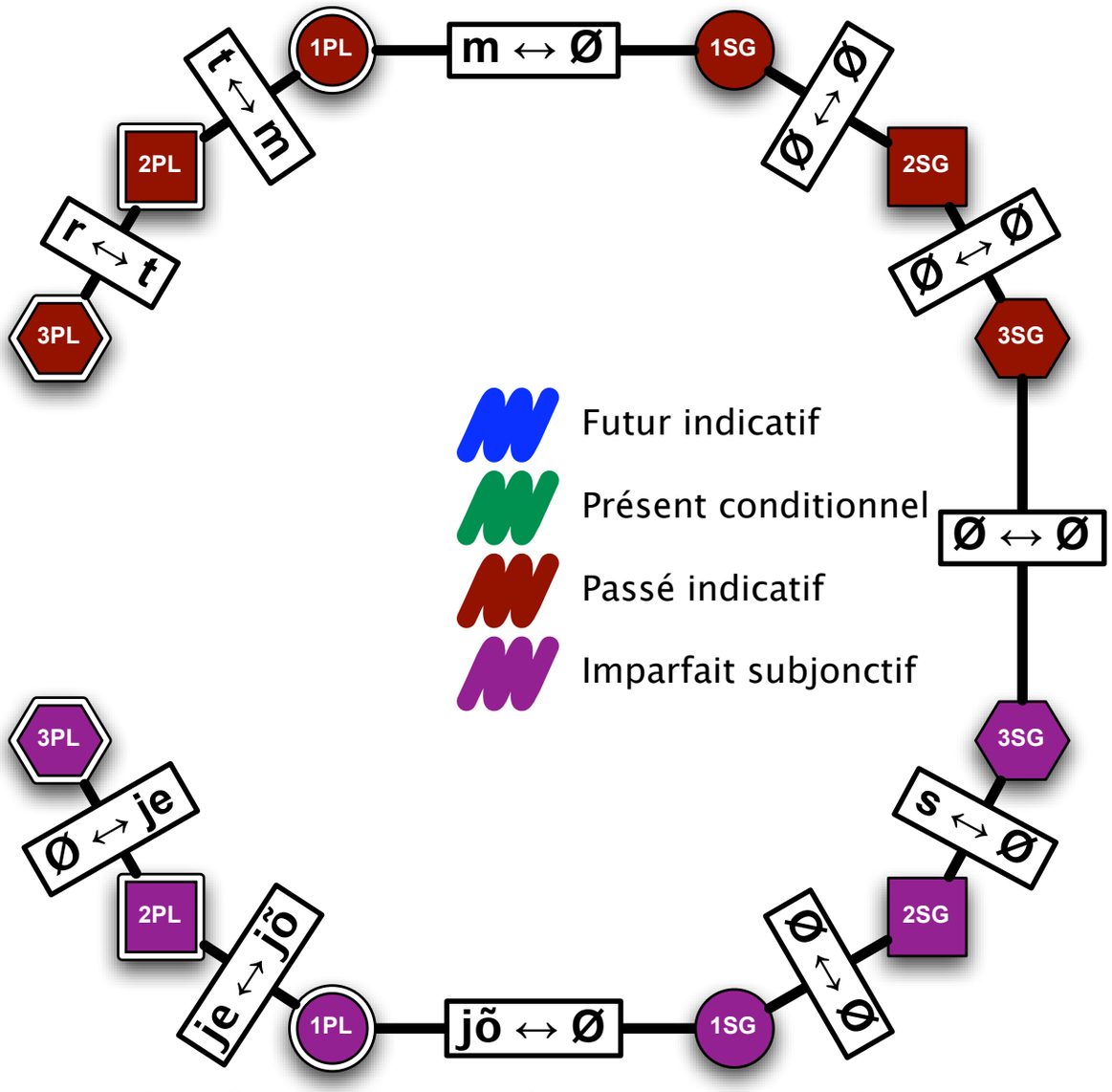
Blocks of forms

- For future and conditional, the 12 forms are inter-predictable for all verbs:
  - 66 symmetric relations link all the forms together
  - 11 relations are sufficient for a linguistic description
    - 239.500.800 possible exhaustive paths
    - all paths are linguistically equivalent
  - the 12 forms constitute a inter-predictable block of inflectional morphology: a block of forms

Links between blocks of forms

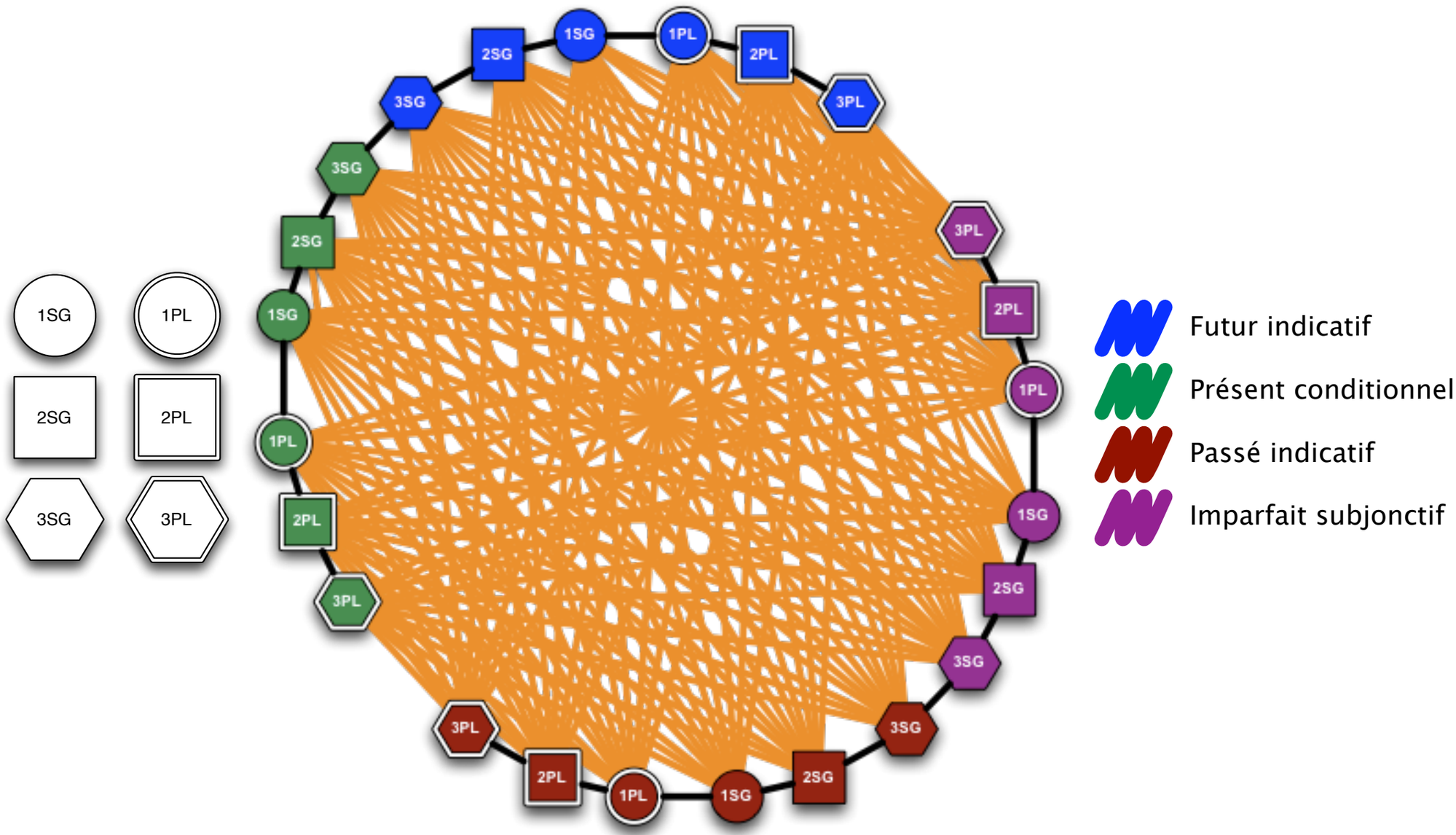
- Between blocks, forms are sometimes inter-predictable
  - we could establish relations between all the forms between blocks
  - but one relation is sufficient





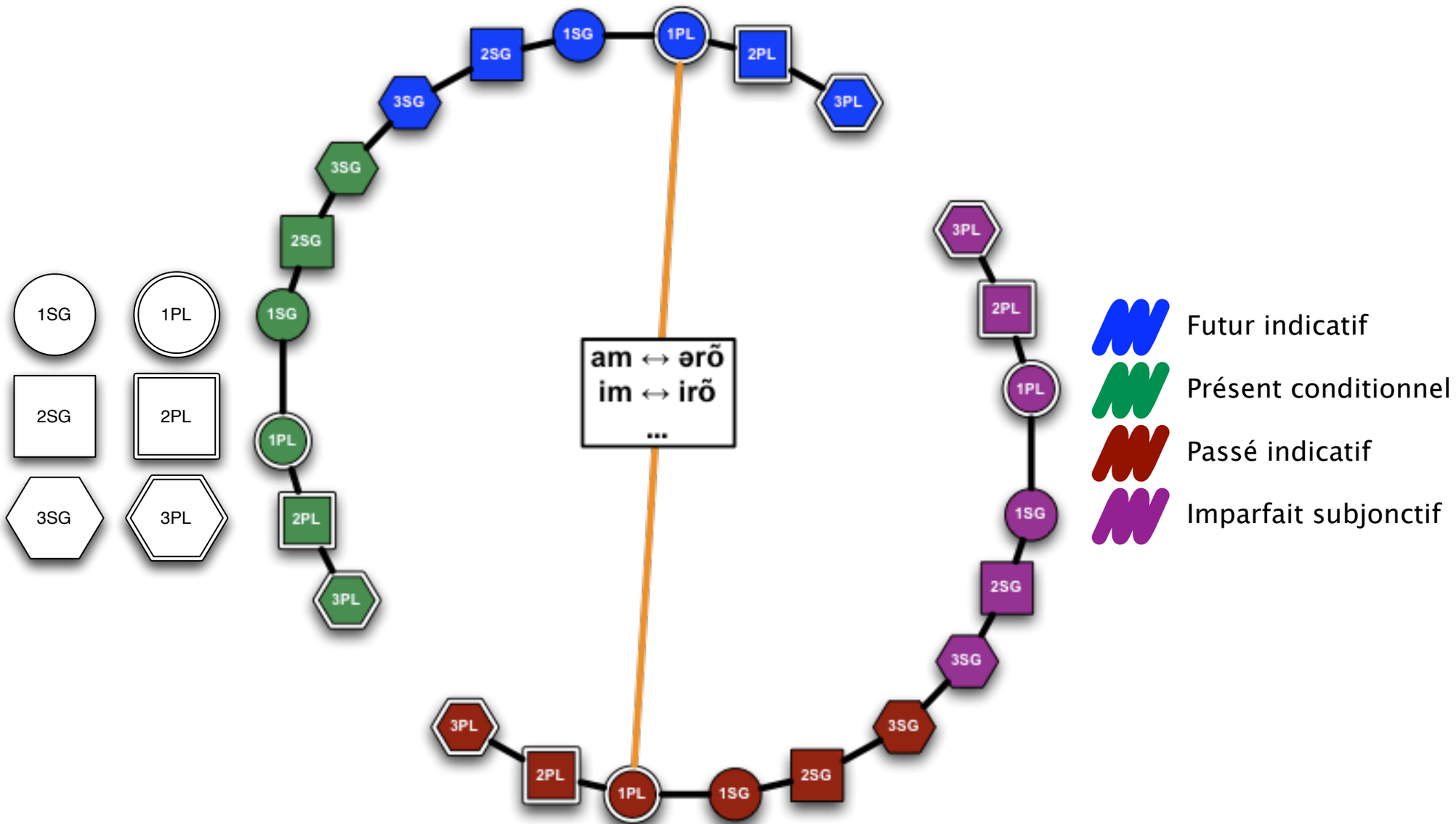
Links between blocks of forms

- Between blocks, forms are sometimes inter-predictable
  - we could establish relations between all the forms between blocks
  - but one relation is sufficient



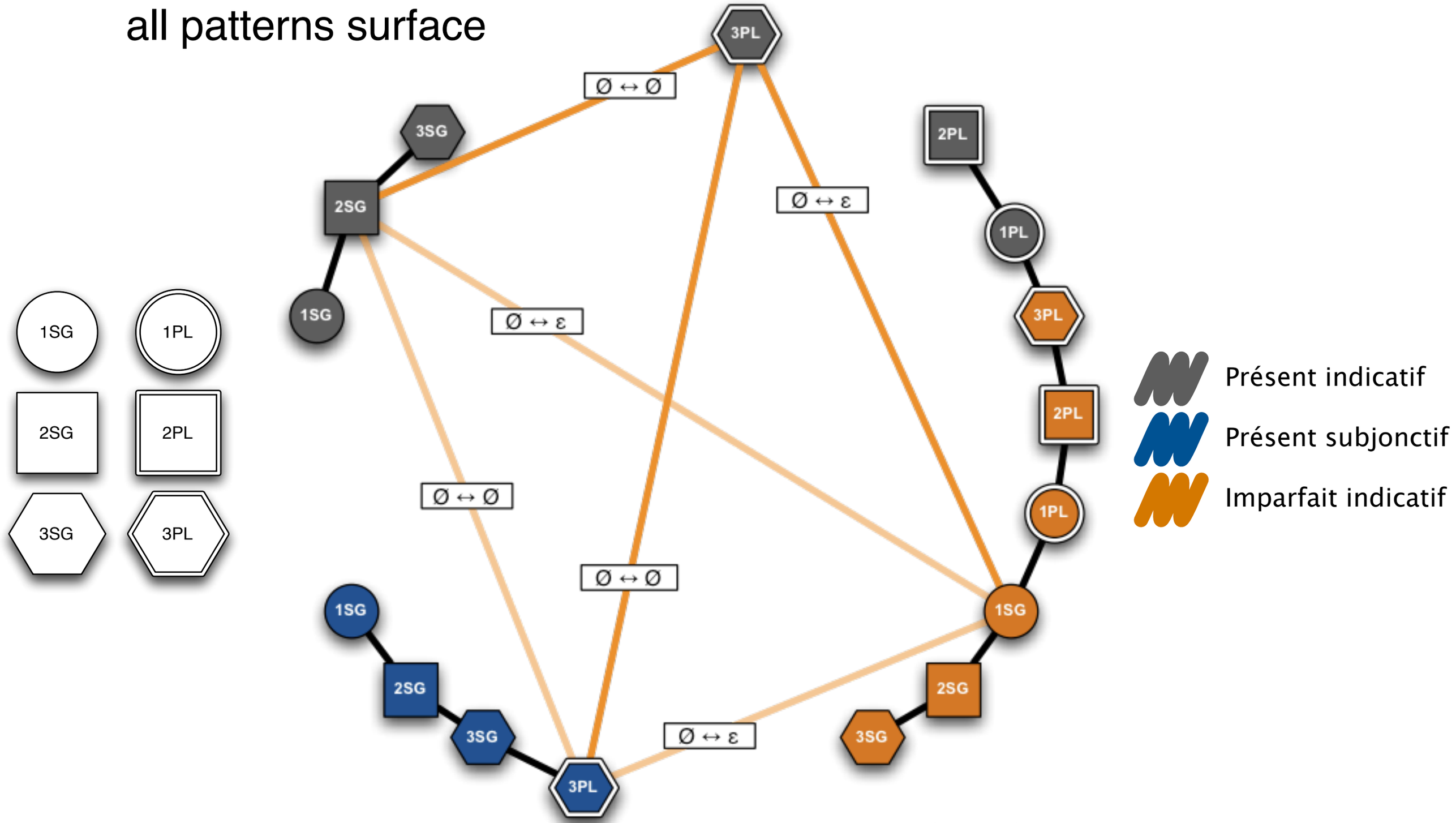
Links between blocks of forms

- Between blocks, forms are sometimes inter-predictable
  - we could establish relations between all the forms between blocks
  - but one relation is sufficient



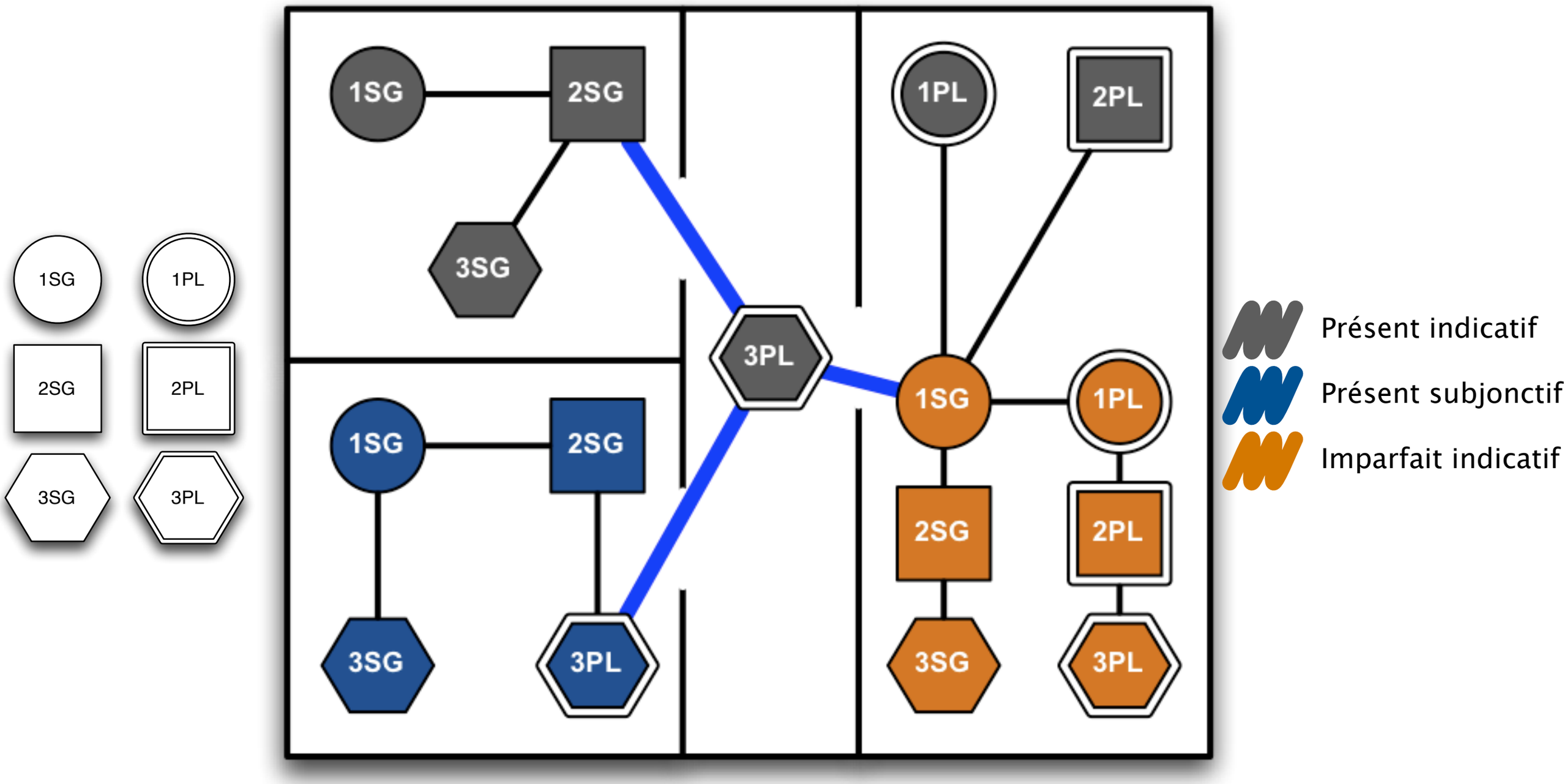
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface



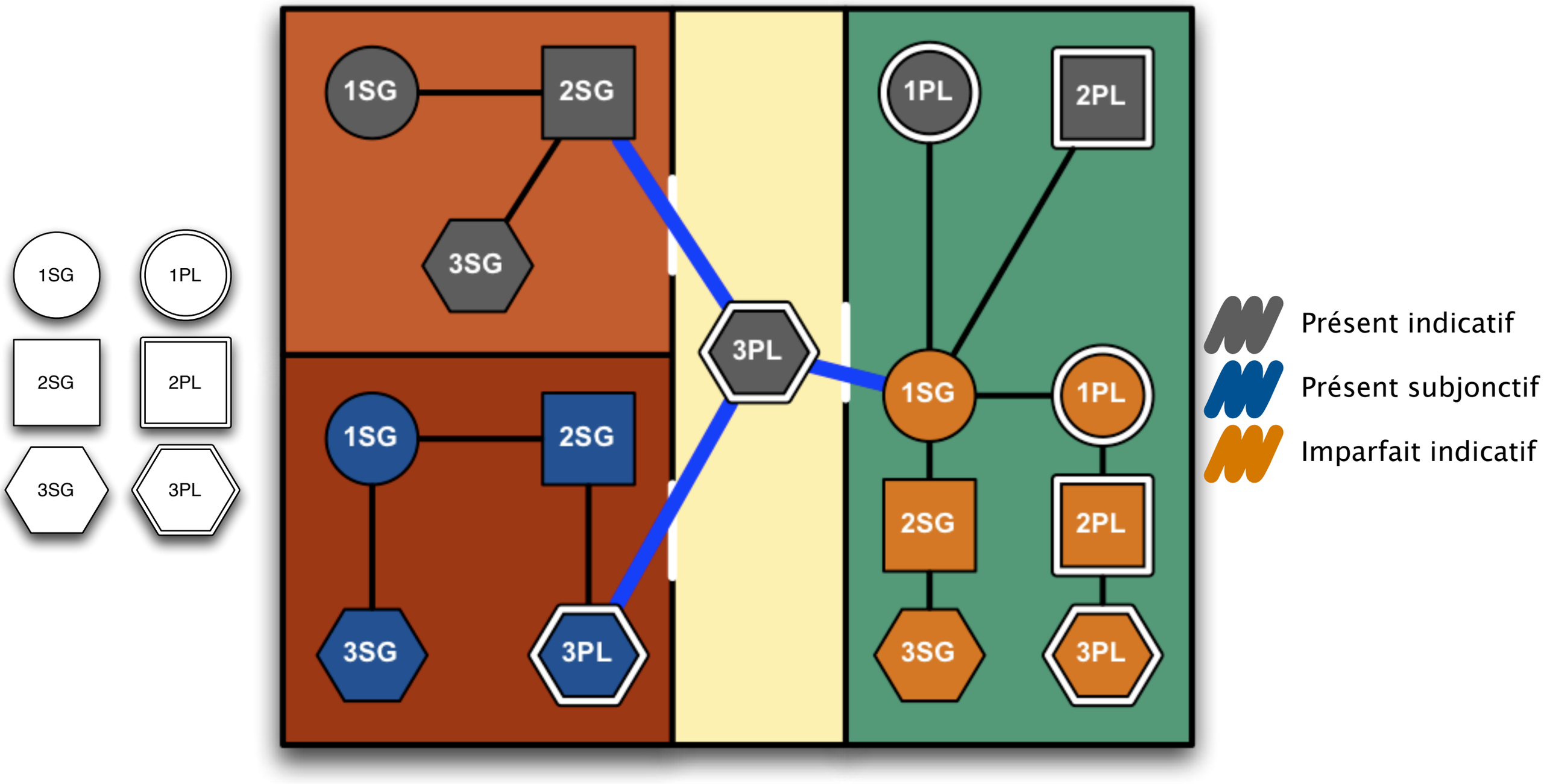
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface



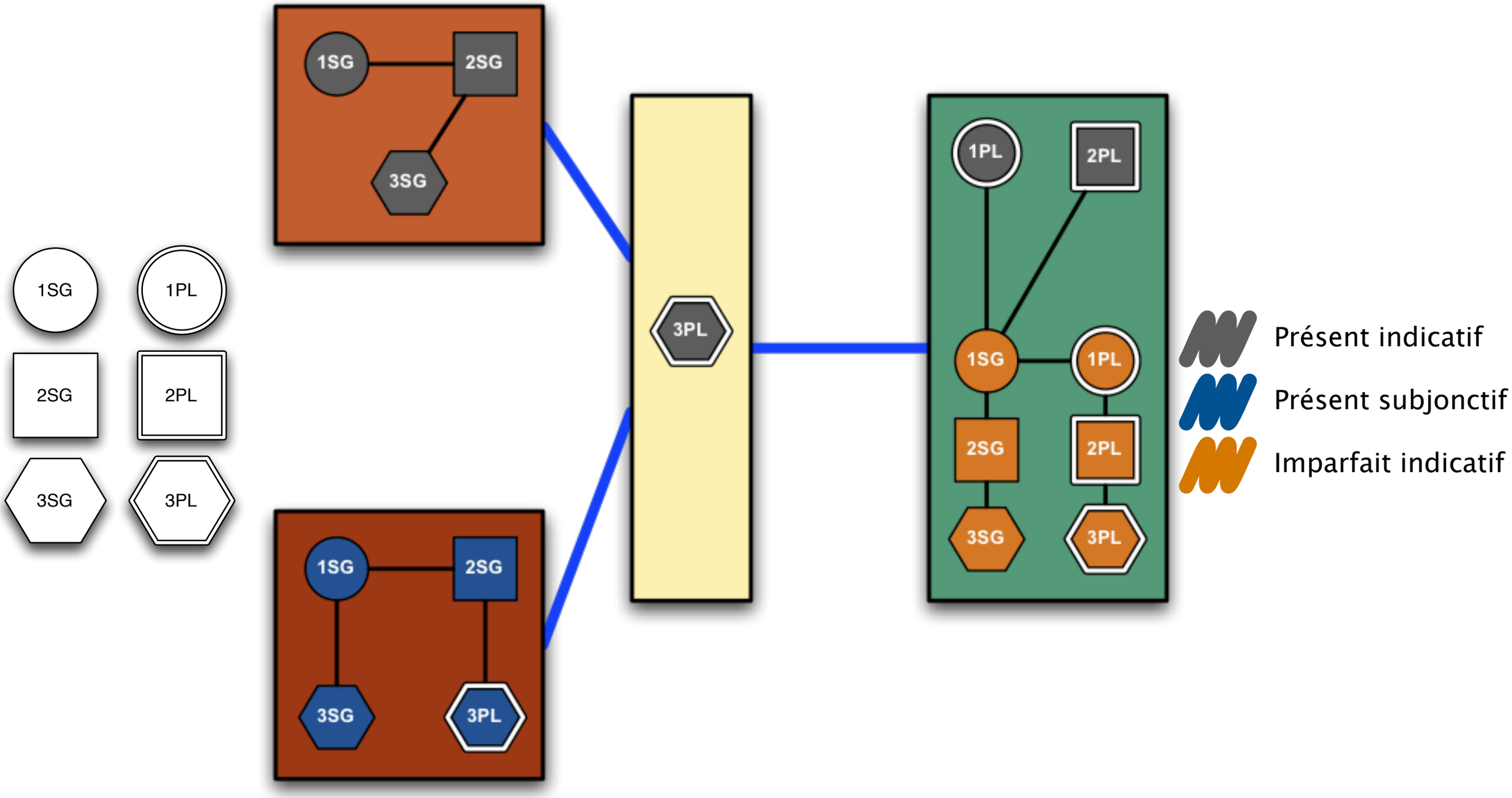
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface



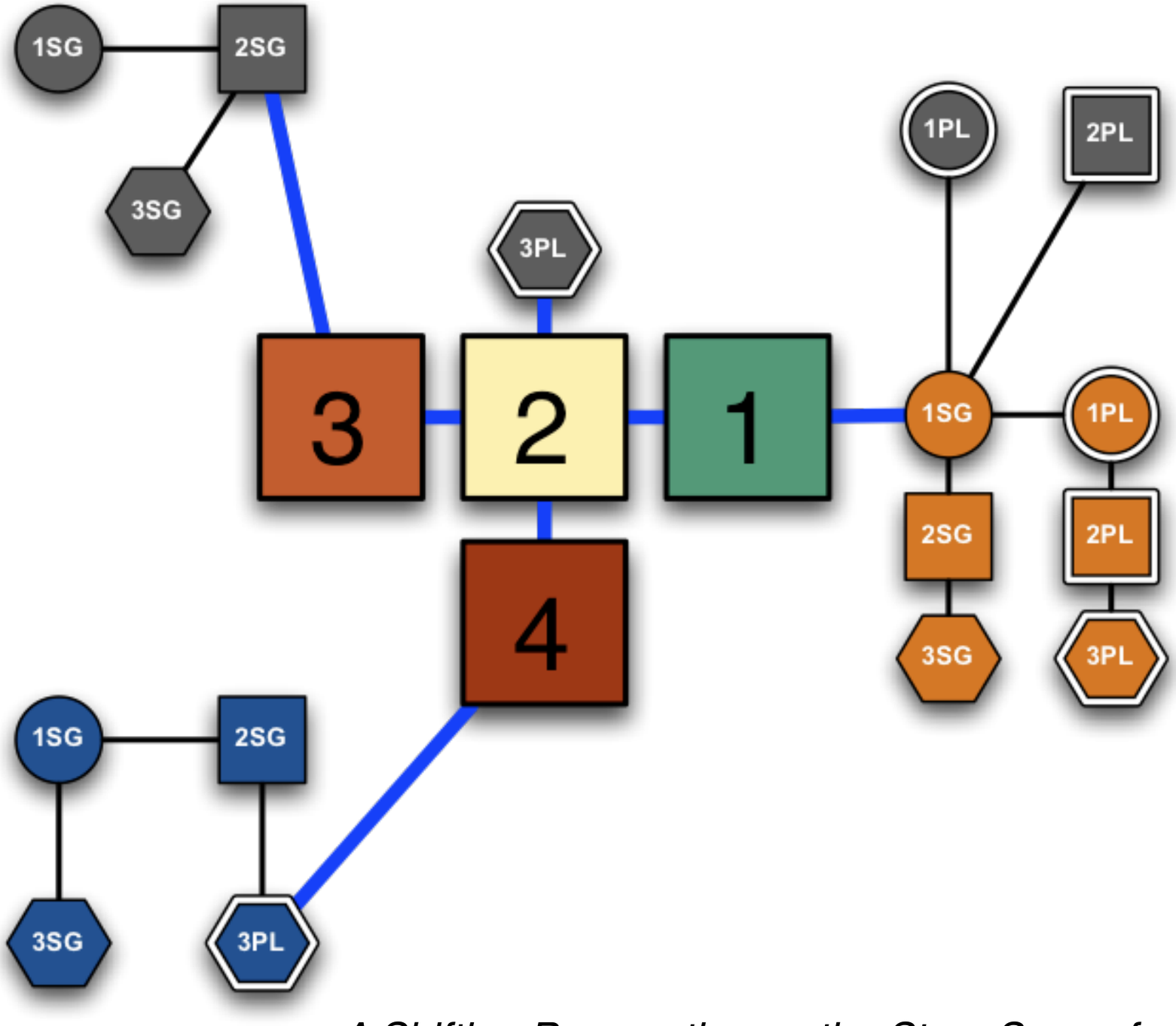
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface



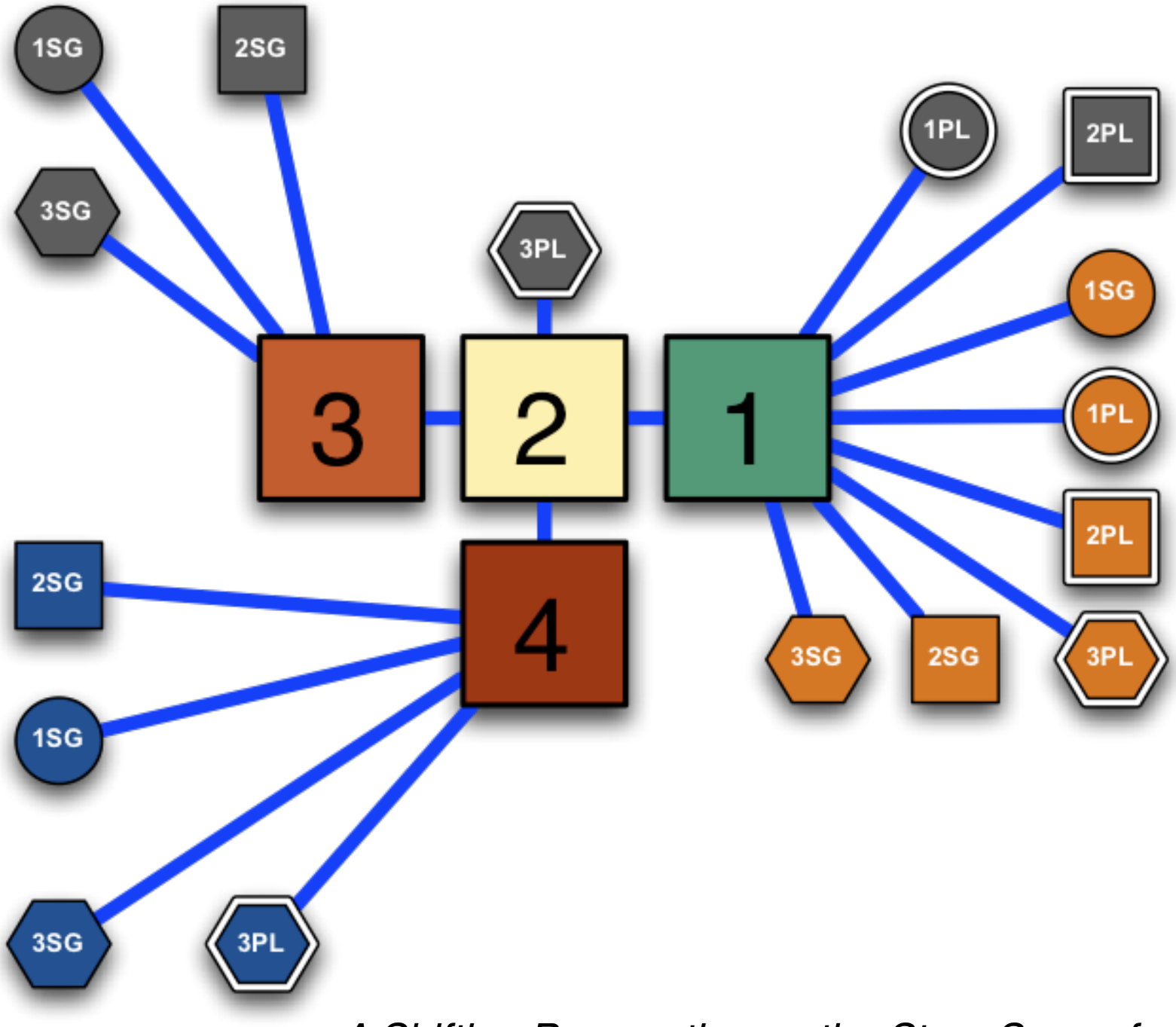
Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface



Irregularity patterns

- For irregular verbs, the distribution of allomorphy is constrained, not all patterns surface



Stem space

- The stem space captures the blocks of forms.
  - The forms in a block are all linked to the same stem.
- The stem graph captures the distribution of semi-regularity.
  - The stem relations constitute an acyclic undirected connected graph (i.e. a tree)
  - The paradigm is minimally connected
  - Heuristics for constructing the stem graph:
    - Stem slots which hold related values when all others are different are linked directly.
    - If all the stem slots are related but one, that stem slot has only one link (tree leaf)

Modelling stem spaces

Two alternatives

Constructive vs. abstractive approaches

- Blevins (2006): distinction between **constructive** and **abstractive** approaches to morphology.
  - In a constructive approach, the basic units are roots (or stems); rules of inflectional morphology specify how exponents are added/ applied to roots/stems to specify words.
  - In an abstractive approach, the basic units are words; rules of inflectional morphology specify how the word filling some cell of the paradigm can be determined on the basis of the content of other cells.
- Blevins's choice of words is somewhat misleading: abstracting away stems (or roots) plays no role in the type of analysis he proposes.
- We suggest that stems can be useful within an abstractive approach.

A semi-constructive approach to stem spaces

- Bonami & Boyé (2006,2007) propose an HPSG approach to French conjugation which is a hybrid between a constructive and abstractive approach:
  - Stem spaces are complex representations within lexical entries, structured by default morphophonological relations.
  - Word forms are deduced from the stem space using a paradigm function (Stump, 2001)

v-lexeme→

$$\begin{bmatrix} \text{HEAD} & \textit{verb} \\ \text{STEMS} & \begin{bmatrix} \textit{v-stem-space} \\ \text{SLOT1} & \textit{phon} \\ \text{SLOT2} & \textit{phon} \\ \text{SLOT3} & \textit{phon} \\ \dots \end{bmatrix} \end{bmatrix}$$

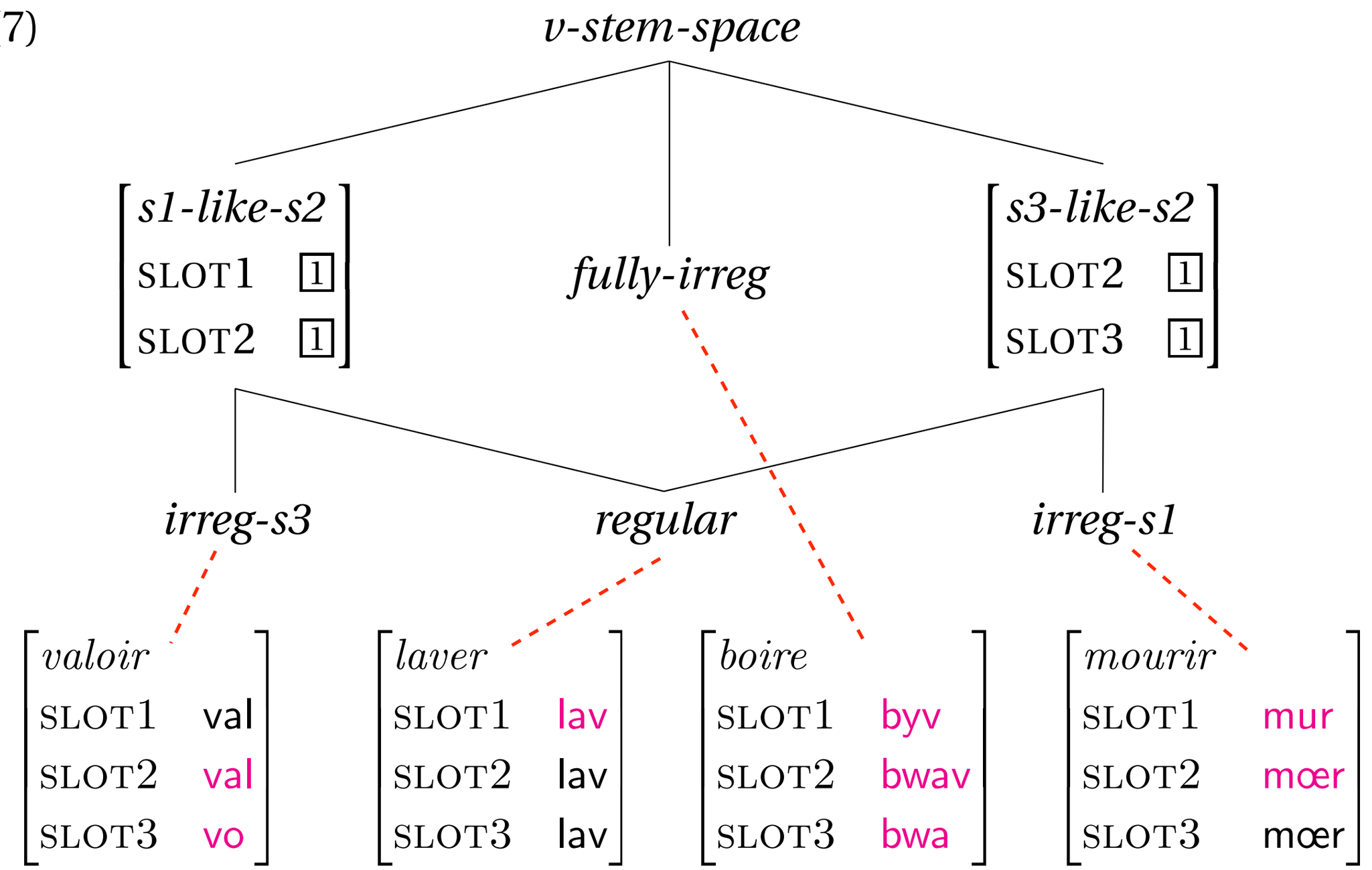
boire:

$$\begin{bmatrix} \text{STEMS} & \begin{bmatrix} \textit{v-stem-space} \\ \text{SLOT1} & \text{byv} \\ \text{SLOT2} & \text{bwav} \\ \text{SLOT3} & \text{bwa} \end{bmatrix} \end{bmatrix}$$

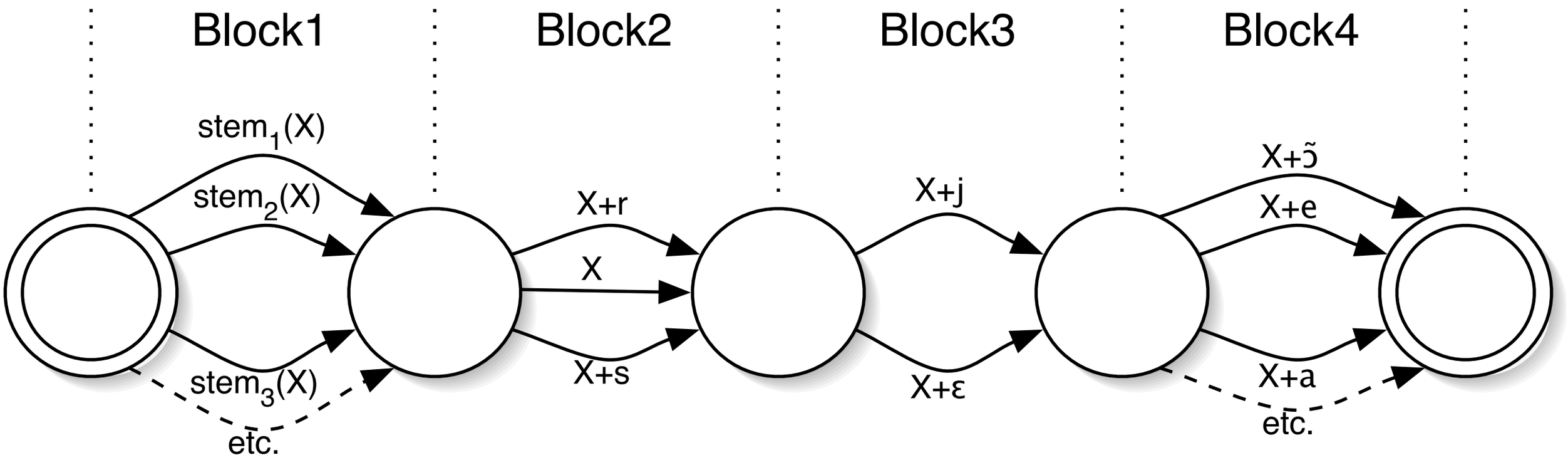
Filling the stem space

(8) *verb-lexeme* → [STEMS / *regular*]

(7)



The PFM component



Block 3

a. $X_V, \sigma : \{\text{RELTYPE } \textit{ana}\} \longrightarrow X \oplus \varepsilon$

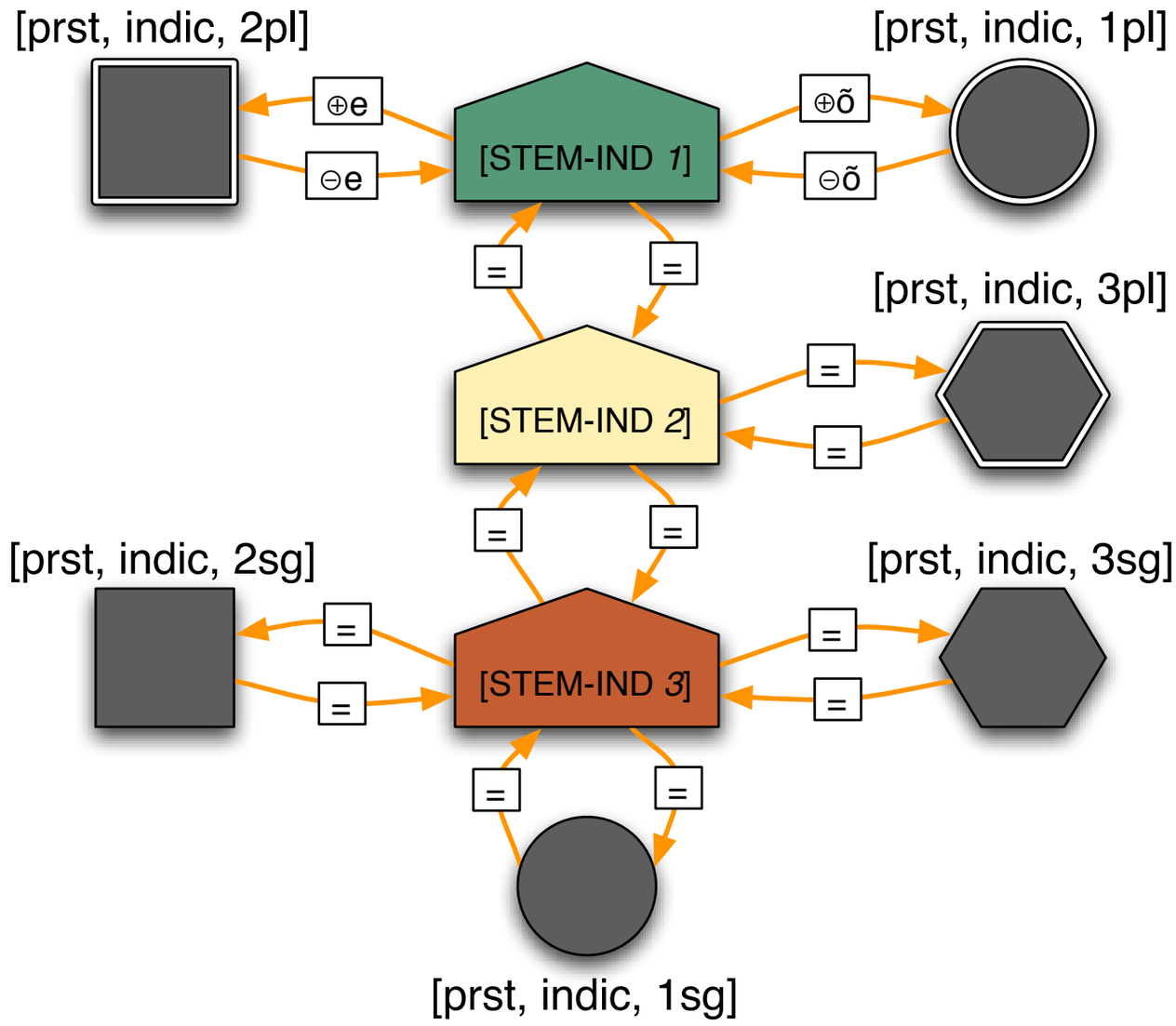
b. $X_V, \sigma : \{\text{RELTYPE } \textit{ana}, \text{ PER } \textit{1}, \text{ NB } \textit{pl}\} \longrightarrow X \oplus j$

c. $X_V, \sigma : \{\text{MODE } \textit{subj}, \text{ PER } \textit{1}, \text{ NB } \textit{pl}\} \longrightarrow X \oplus j$

d. $X_V, \sigma : \{\text{PER } \textit{2}, \text{ NB } \textit{pl}\} \longrightarrow \langle X, \sigma / \{\text{PER } \textit{1}\} \rangle : 3$

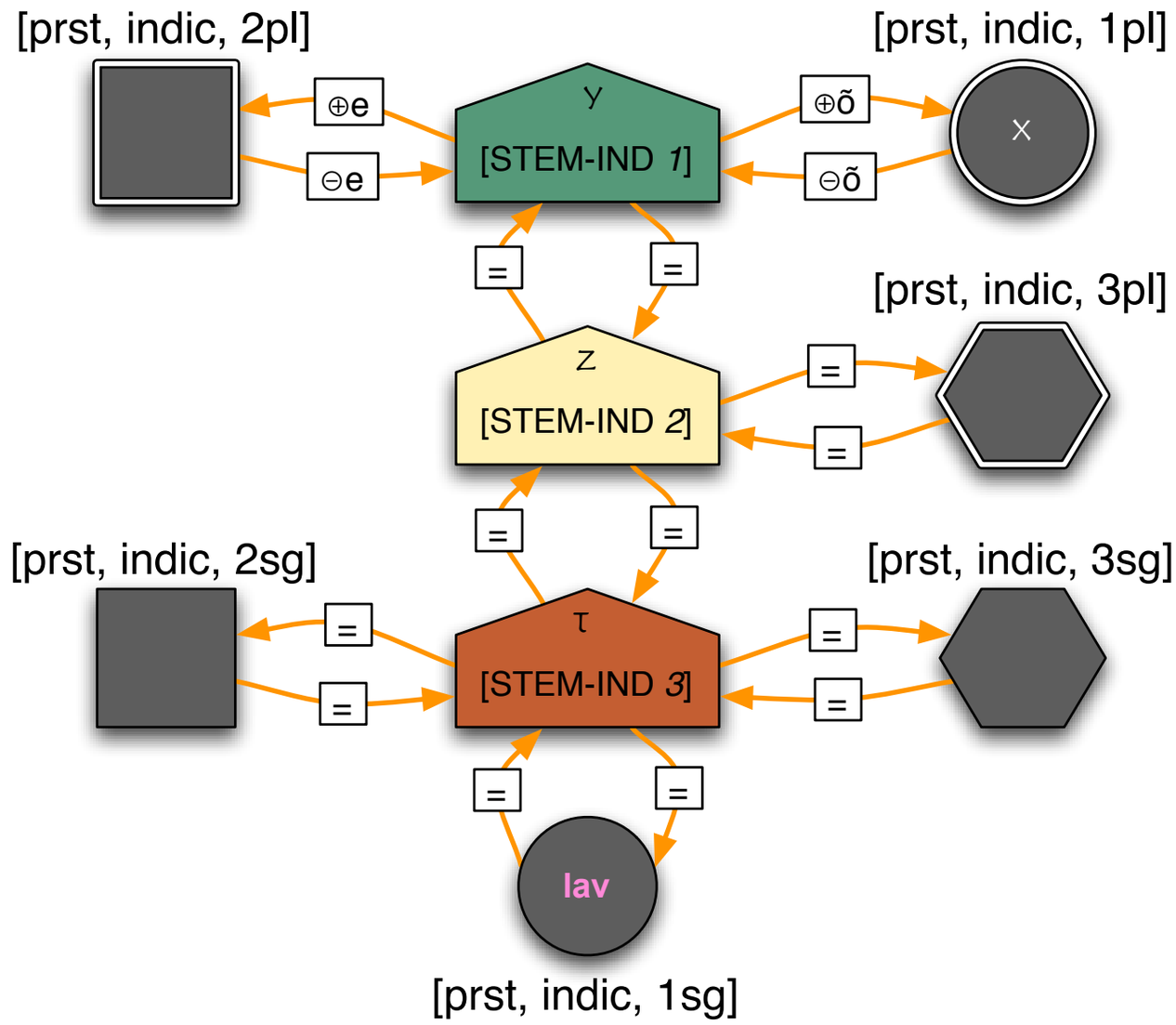
A fully abstractive alternative

- Come back to the view of inflection as an oriented graph
  - Arcs within the graph can be modelled by standard lexical rules
  - Stems are just nodes with no morphosyntactic description
  - See e.g. the indicative present:



A fully abstractive alternative

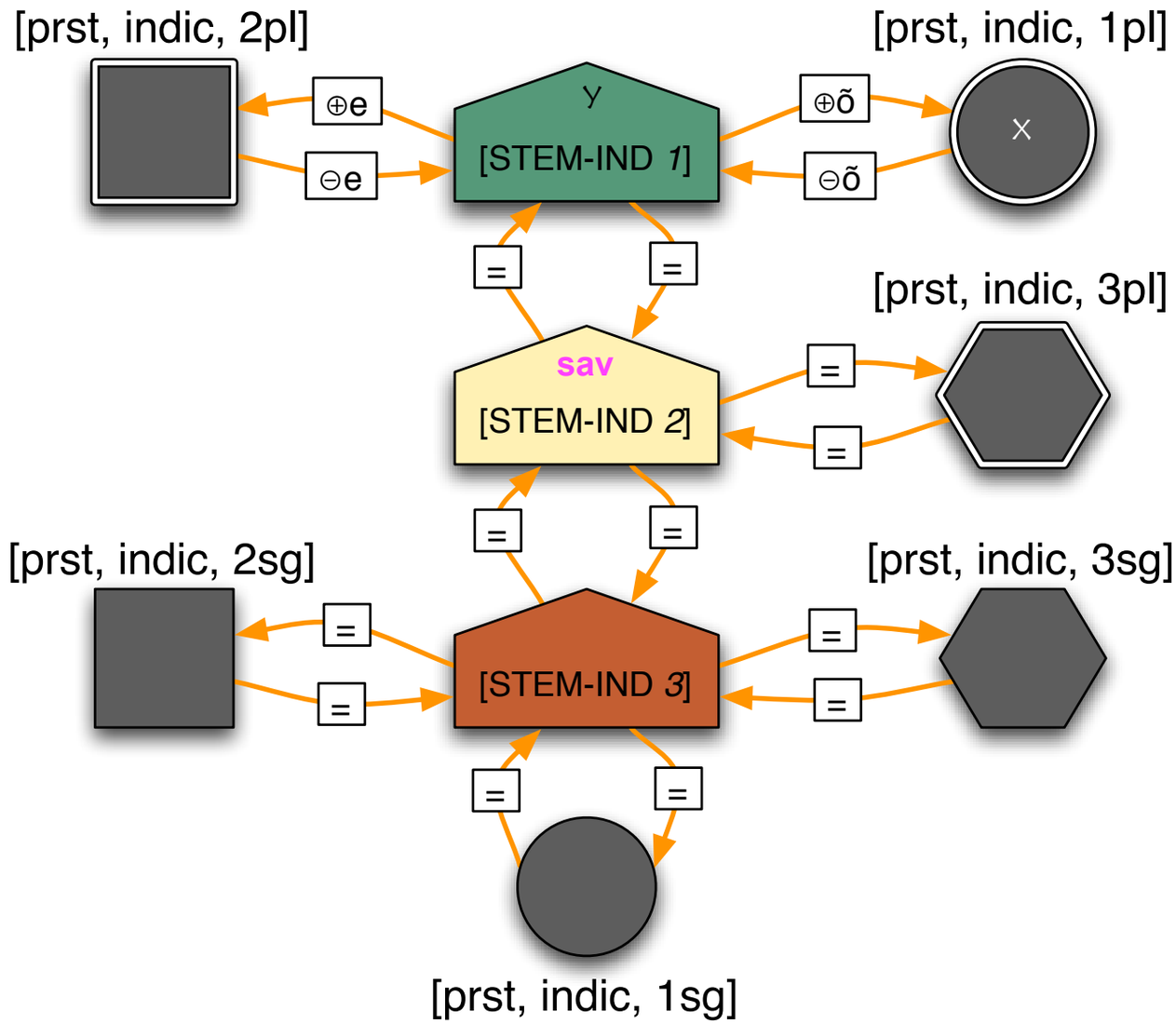
- Determining the form filling a slot amounts to circulating the graph until one finds a lexical entry
  - To inflect correctly *laver* in the 1pl, the speaker can rely on the kwowledge of any other form or stem. Suppose he knows only the 1sg :



τ = lav
z = τ = lav
y = z = lav
x = y⊕õ = lavõ

A fully abstractive alternative

- Determining the form filling a slot amounts to circulating the graph until one finds a lexical entry
  - To inflect correctly *savoir* in the 1pl, the speaker needs to know more specific information—e.g. stem 2.



HPSG implementation: constraints on types

- Lexemes do not exist as data structures. Rather, a collection of lexical objects share a *lexemic index* (*à la* Spencer)

$$\textit{lex-obj} \rightarrow \left[\text{LXM} \quad \textit{lxm-ind}\right]$$

- Two types of lexical objects:
  - Inflectional relations specify how some form of a lexeme depends on another form of the same lexeme

$$\textit{infl-rln} \rightarrow \left[\begin{array}{ll}\text{LXM} & \boxed{1} \\ \text{DTRS} & \left\langle \left[\begin{array}{l}\textit{lex-obj} \\ \text{LXM } \boxed{1}\end{array}\right] \right\rangle\end{array}\right]$$

  - Lexical entries are lexical objects that do not depend on any other lexical object

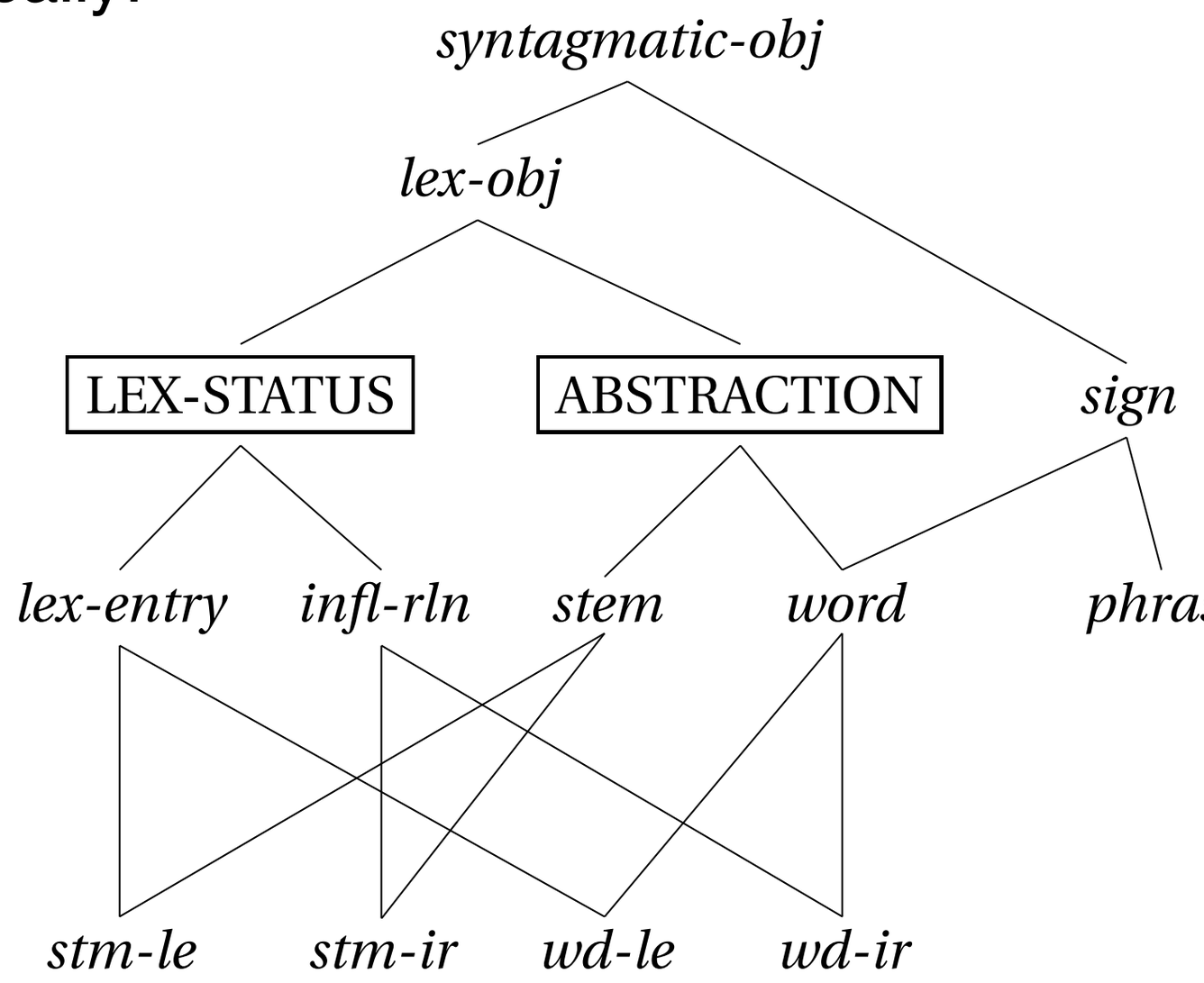
$$\textit{lex-entry} \rightarrow \left[\text{DTRS} \quad \langle\rangle\right]$$

- Lexical entries win over inflectional relations

$$\textit{lex-obj} \rightarrow (\textit{lex-entry} \vee (\neg \textit{lex-entry} \wedge \textit{infl-rln}))$$

HPSG implementation: the type hierarchy

- More technically:



$$syntagmatic\text{-}obj \rightarrow \left[\text{PHON} \quad phon\right] \qquad sign \rightarrow \left[\text{SYNSEM} \quad synsem\right]$$

HPSG implementation: the network

- Concentrating again on the present indicative:

infl-rln→

$$\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{SYNSEM} & [sg] \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 3\end{array}\right]\right\rangle\end{array}\right] \vee \left[\begin{array}{ll}\text{PHON} & \boxed{1}\oplus\tilde{\text{ɔ}} \\ \text{SS} & [1pl] \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 1\end{array}\right]\right\rangle\end{array}\right] \vee \left[\begin{array}{ll}\text{PHON} & \boxed{1}\oplus\text{e} \\ \text{SS} & [2pl] \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 1\end{array}\right]\right\rangle\end{array}\right] \vee$$

$$\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{SS} & [3pl] \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 2\end{array}\right]\right\rangle\end{array}\right] \vee \left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 3 \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{SS} & [sg]\end{array}\right]\right\rangle\end{array}\right] \vee \left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 1 \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1}\oplus\tilde{\text{ɔ}} \\ \text{SS} & [1pl]\end{array}\right]\right\rangle\end{array}\right] \vee$$

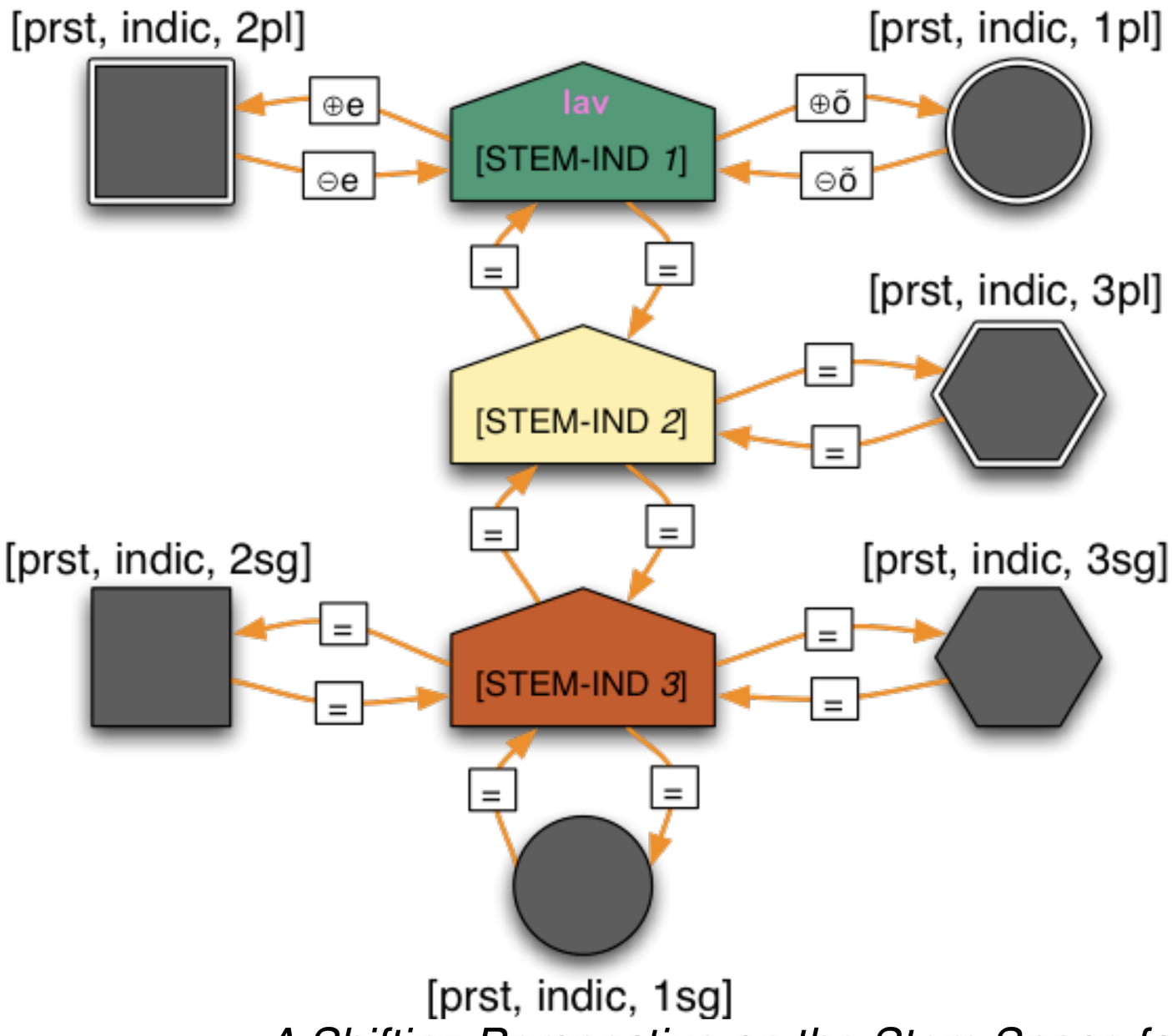
$$\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 1 \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1}\oplus\text{e} \\ \text{SS} & [2pl]\end{array}\right]\right\rangle\end{array}\right] \vee \left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 2 \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{SS} & [3pl]\end{array}\right]\right\rangle\end{array}\right] \vee \left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 3 \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 2\end{array}\right]\right\rangle\end{array}\right] \vee$$

$$\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 2 \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 3\end{array}\right]\right\rangle\end{array}\right] \vee \left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 1 \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 2\end{array}\right]\right\rangle\end{array}\right] \vee \left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 2 \\ \text{DTRS} & \left\langle\left[\begin{array}{ll}\text{PHON} & \boxed{1} \\ \text{STM-IND} & 1\end{array}\right]\right\rangle\end{array}\right]$$

HPSG implementation: the lexical entries

- Minimal specifications for some lexemes:

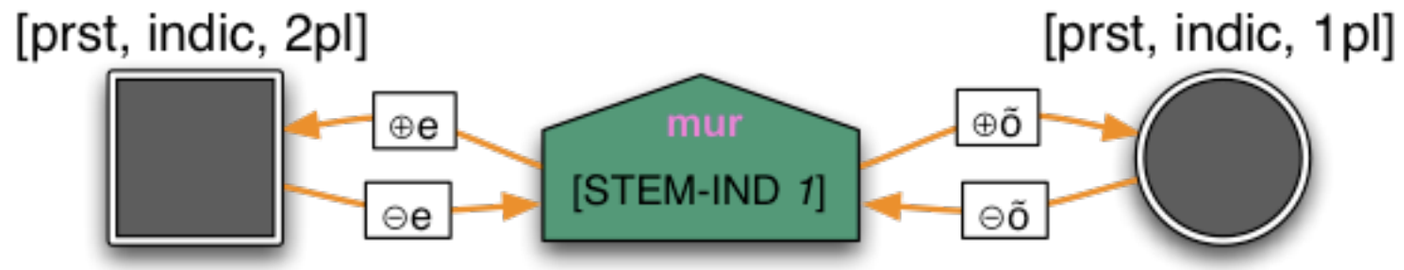
$$\begin{bmatrix} \textit{lex-entry} & \\ \text{LXM} & \textit{laver} \end{bmatrix} \rightarrow \begin{bmatrix} \text{PHON} & \text{lav} \\ \text{STM-IND} & \textit{1} \end{bmatrix}$$



HPSG implementation: the lexical entries

- Minimal specifications for some lexemes:

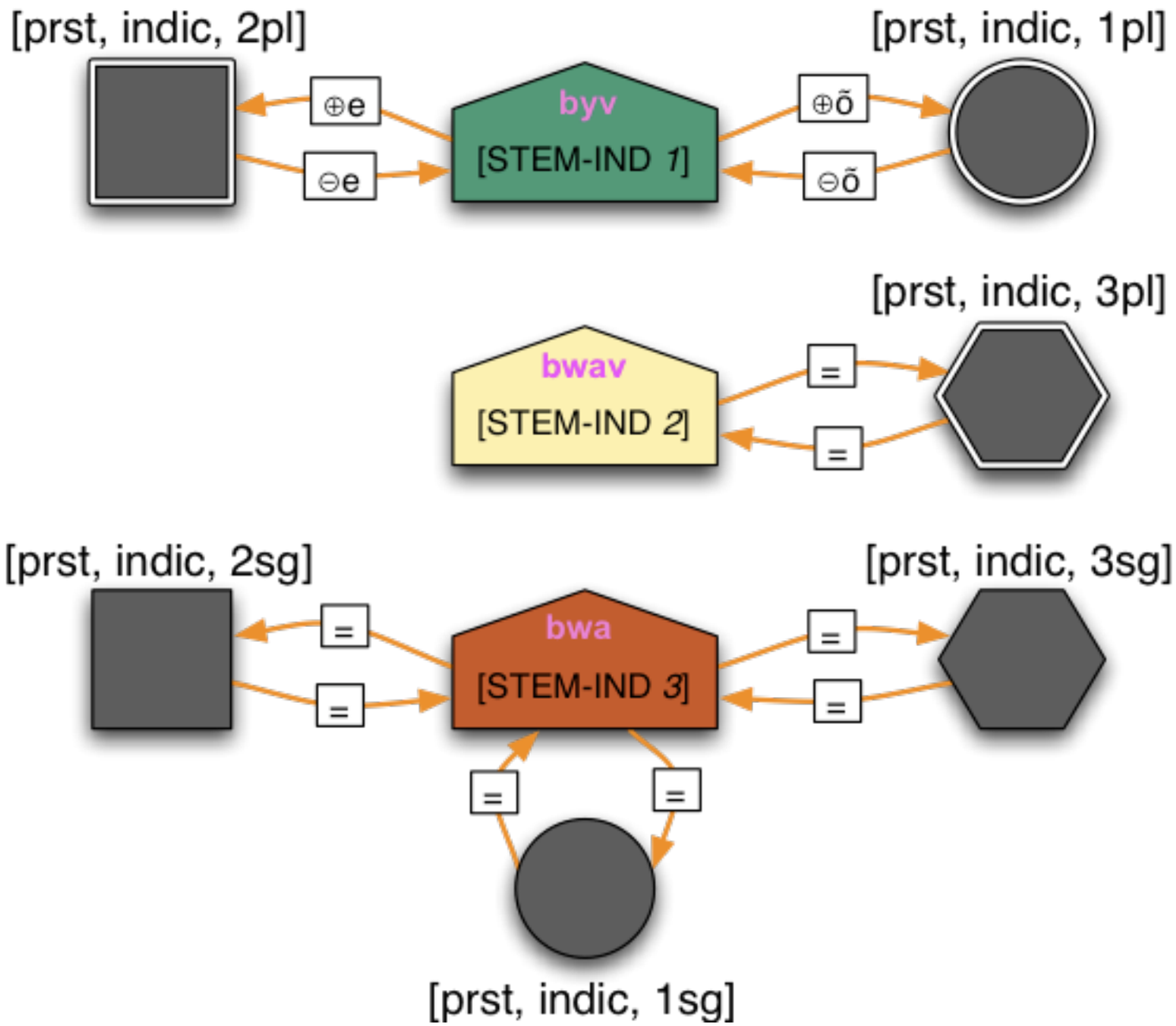
$$\begin{bmatrix} \textit{lex-entry} & \\ \text{LXM} & \textit{mourir} \end{bmatrix} \rightarrow \begin{bmatrix} \text{PHON} & \text{mur} \\ \text{STM-IND} & \textit{1} \end{bmatrix} \vee \begin{bmatrix} \text{PHON} & \text{mœr} \\ \text{STM-IND} & \textit{2} \end{bmatrix}$$



HPSG implementation: the lexical entries

- Minimal specifications for some lexemes:

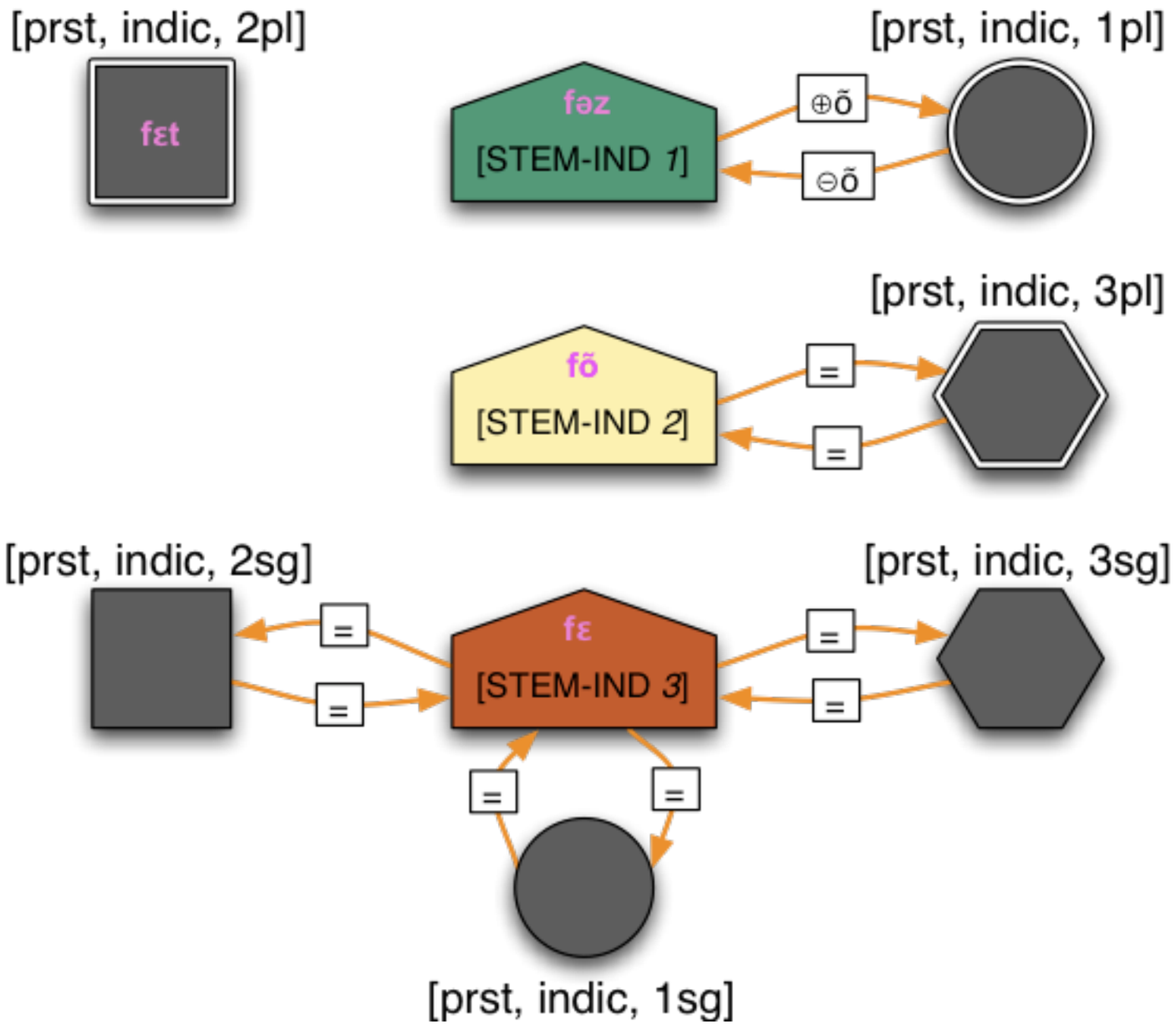
$$\begin{bmatrix} \textit{lex-entry} & \\ \text{LXM} & \textit{boire} \end{bmatrix} \rightarrow \begin{bmatrix} \text{PHON} & \text{byv} \\ \text{STM-IND} & 1 \end{bmatrix} \vee \begin{bmatrix} \text{PHON} & \text{bwav} \\ \text{STM-IND} & 2 \end{bmatrix} \vee \begin{bmatrix} \text{PHON} & \text{bwa} \\ \text{STM-IND} & 3 \end{bmatrix}$$



HPSG implementation: the lexical entries

- Minimal specifications for some lexemes:

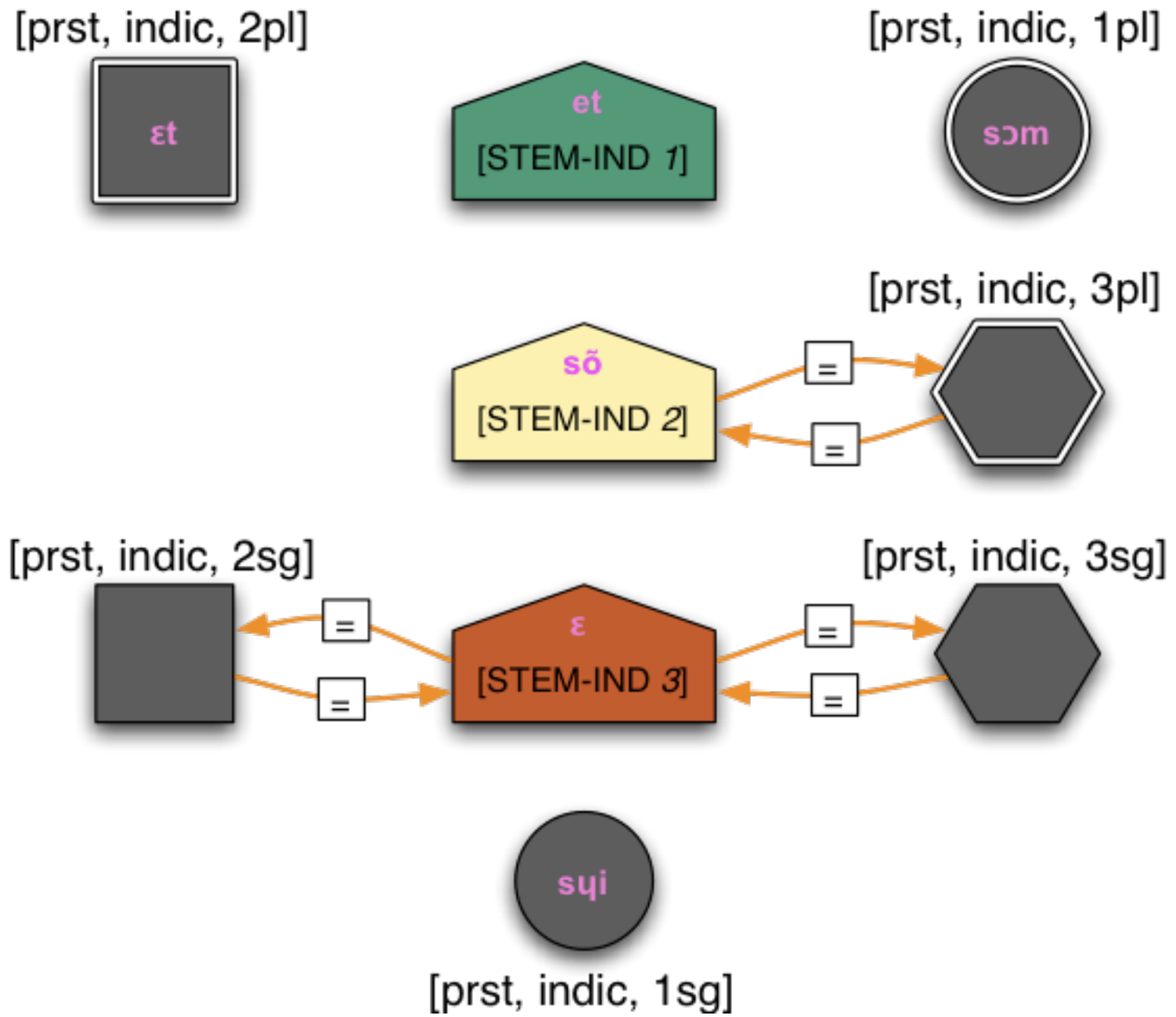
$$\begin{bmatrix} \textit{lex-entry} \\ \text{LXM} \quad \textit{faire} \end{bmatrix} \rightarrow \begin{bmatrix} \text{PHON} & \text{fəz} \\ \text{STM-IND} & \textit{1} \end{bmatrix} \vee \begin{bmatrix} \text{PHON} & \text{fɔ̃} \\ \text{STM-IND} & \textit{2} \end{bmatrix} \vee \begin{bmatrix} \text{PHON} & \text{fɛ} \\ \text{STM-IND} & \textit{3} \end{bmatrix} \vee \begin{bmatrix} \text{PHON} & \text{fɛt} \\ \text{SS} & [\textit{2pl}] \end{bmatrix}$$



HPSG implementation: the lexical entries

- Minimal specifications for some lexemes:

$$\begin{bmatrix} \textit{lex-entry} \\ \text{LXM} \quad \textit{être} \end{bmatrix} \rightarrow \begin{bmatrix} \text{PHON} & \text{ɛt} \\ \text{STM-IND} & \textit{1} \end{bmatrix} \vee \begin{bmatrix} \text{PHON} & \text{sɔ̃} \\ \text{STM-IND} & \textit{2} \end{bmatrix} \vee \begin{bmatrix} \text{PHON} & \text{ɛ} \\ \text{STM-IND} & \textit{3} \end{bmatrix} \vee \begin{bmatrix} \text{PHON} & \text{sɥi} \\ \text{SS} & [\textit{1sg}] \end{bmatrix} \vee \begin{bmatrix} \text{PHON} & \text{sɔm} \\ \text{SS} & [\textit{1pl}] \end{bmatrix}$$



The abstractive approach: evaluation

- Shows how an abstractive approach can accomodate stem spaces
- From a technical, HPSG point of view:
  - Captures the relevant generalizations without appealing to disputed apparatus (online type construction, defaults)
- Psycholinguistically realistic:
  - The amount of stored lexical knowledge for two lexemes in the same class need not be the same
  - Only principal parts are needed, but more can be stored.
- Linguistically minimal:
  - Only with very few lexemes is it necessary to store redundant information
  - These correspond roughly to the *complex dependency relations* left aside in Bonami & Boyé 2002.

The abstractive approach: evaluation

- Exponence issue:
  - We have no principled decomposition of exponence *à la* PFM. All endings are atomic.
  - But in fact we do not loose much:
    - the PFM analysis in Bonami & Boyé (2007a) uses 19 rules of exponence + 13 rules of stem selection to generate 48 forms.
    - here we use 48 rules which deal simultaneously with exponence and stem selection
- Redundancy issue:
  - We need to store converse inflectional relations systematically, which seems redundant.
  - as it turns out, the relevant relations are often not really converses… see section 3

Extensions

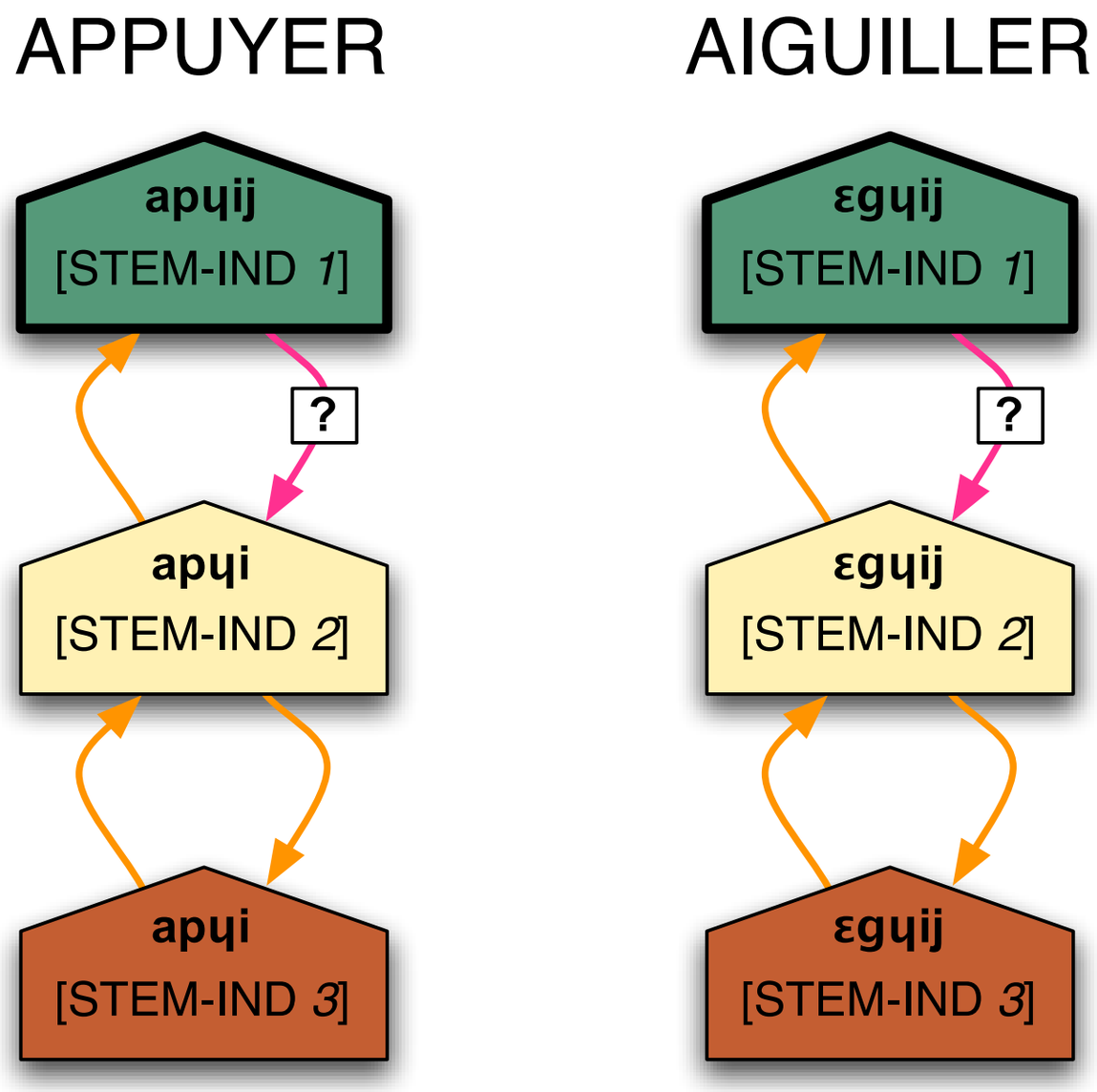
Problem 1: converse opacities

- Assume that regular conjugation patterns in French correspond to the 1st and 2nd traditional groups (Bonami, Boyé, Giraudo & Voga, 2008)
- Then there are at least 6 distinct regular patterns

	PRST 1SG	PRST 2SG	PRST 3SG	PRST 1PL	PRST 2PL	PRST 3PL	INF
APPUYER	apɥi	apɥi	apɥi	apɥij	apɥij	apɥi	apɥije
AIGUILLER	ɛgɥij	ɛgɥij	ɛgɥij	ɛgɥij	ɛgɥij	ɛgɥij	ɛgɥije
ABOYER	abwa	abwa	abwa	abwaj	abwaj	abwa	abwaje
CRÉER	kre	kre	kre	kre	kre	kre	kree
TAPISSER	tapis	tapis	tapis	tapis	tapis	tapis	tapise
TAPIR	tapi	tapi	tapi	tapis	tapis	tapis	tapir

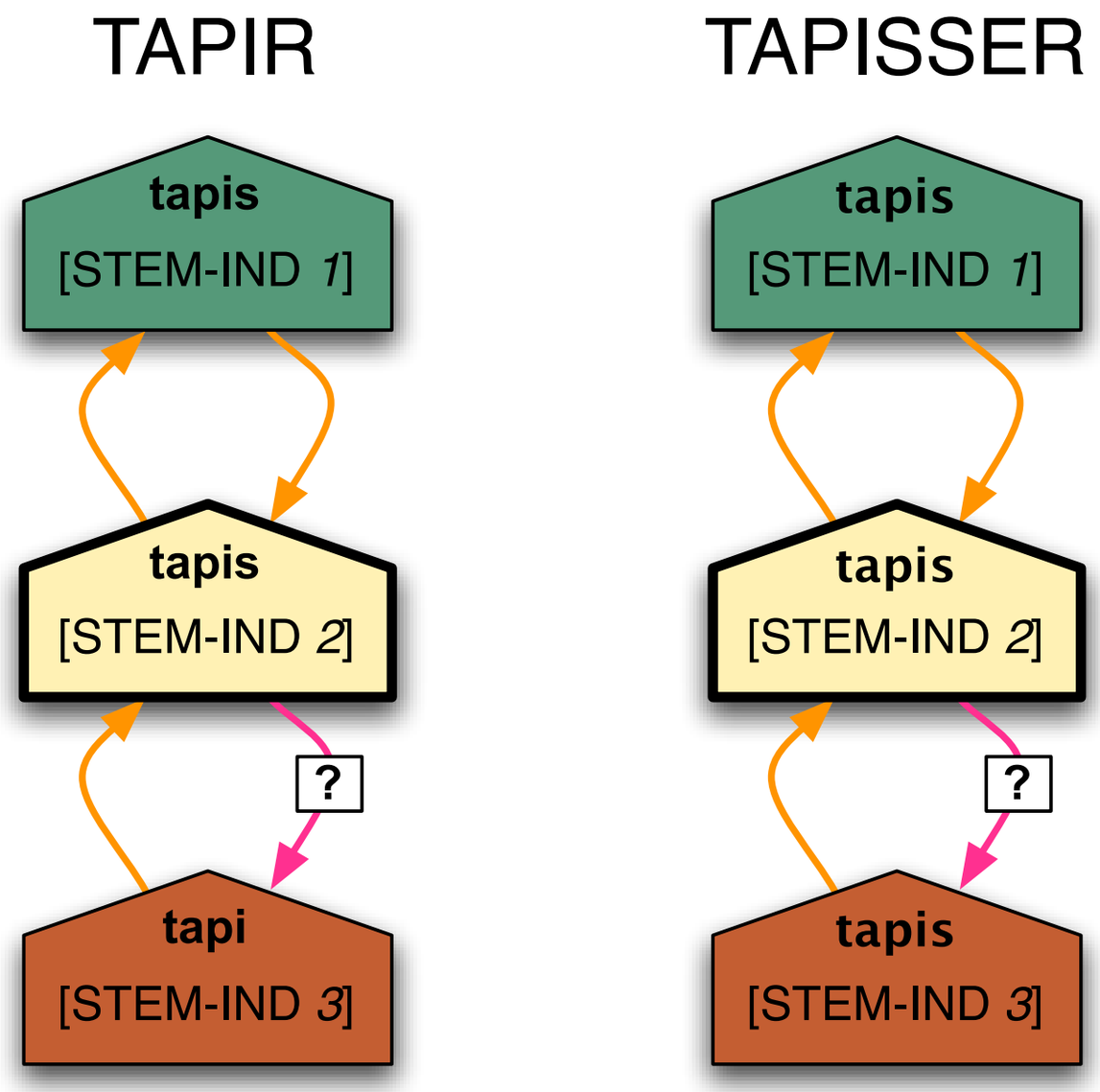
Problem 1: converse opacities

- For some verbs, prediction from stem 1 does not work



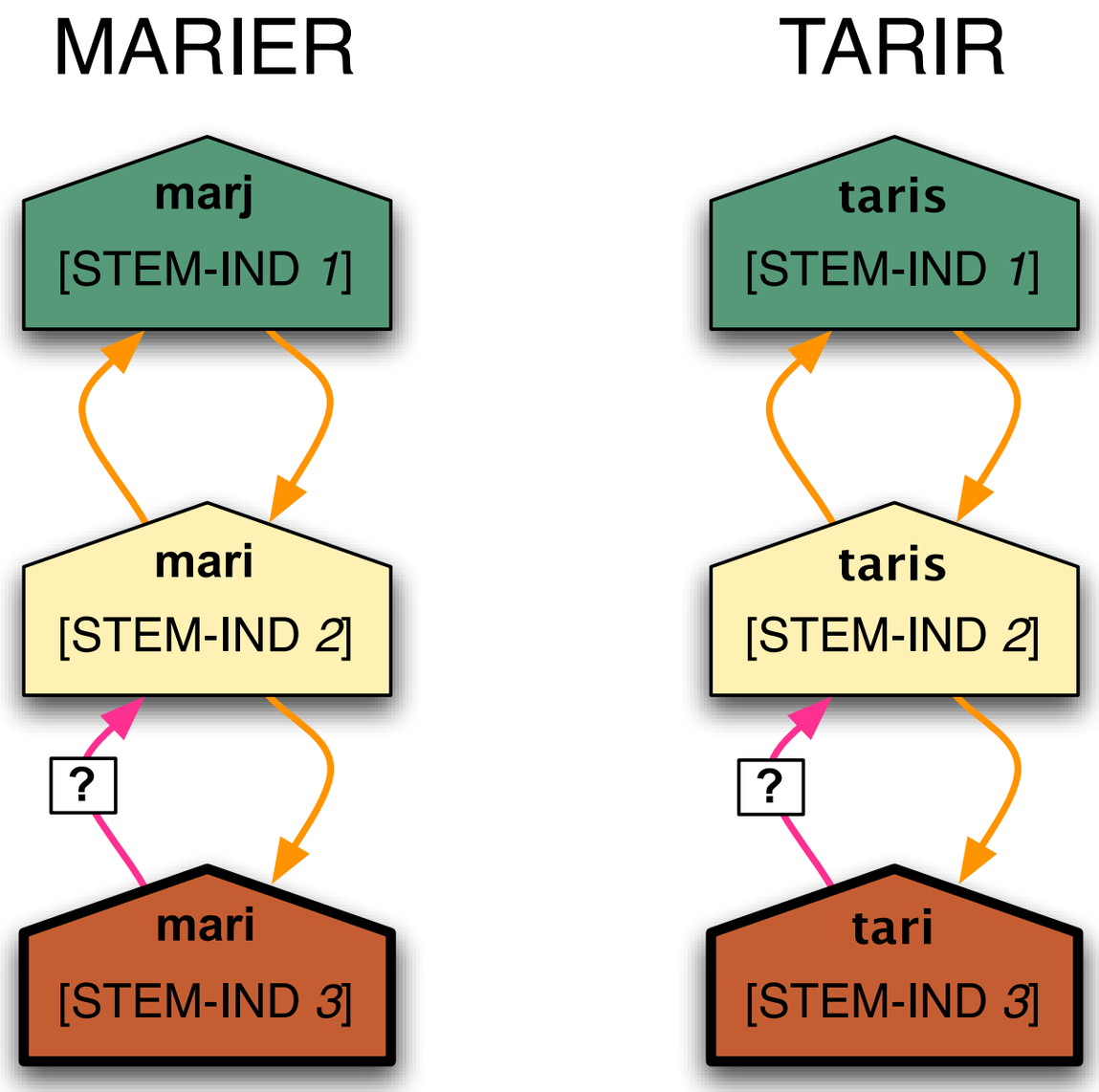
Problem 1: converse opacities

- Neither does stem 2



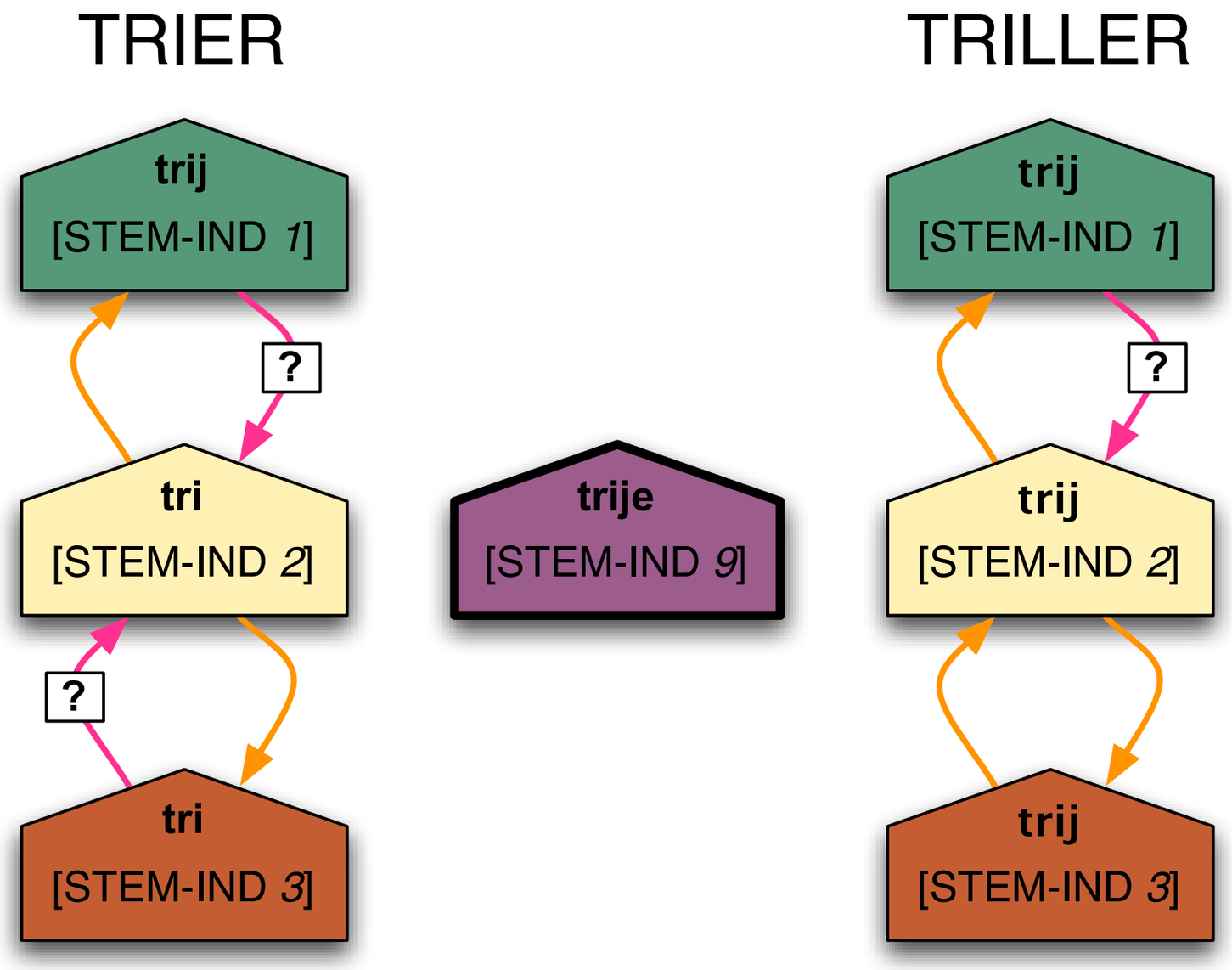
Problem 1: converse opacities

- Stem 3 does not work either



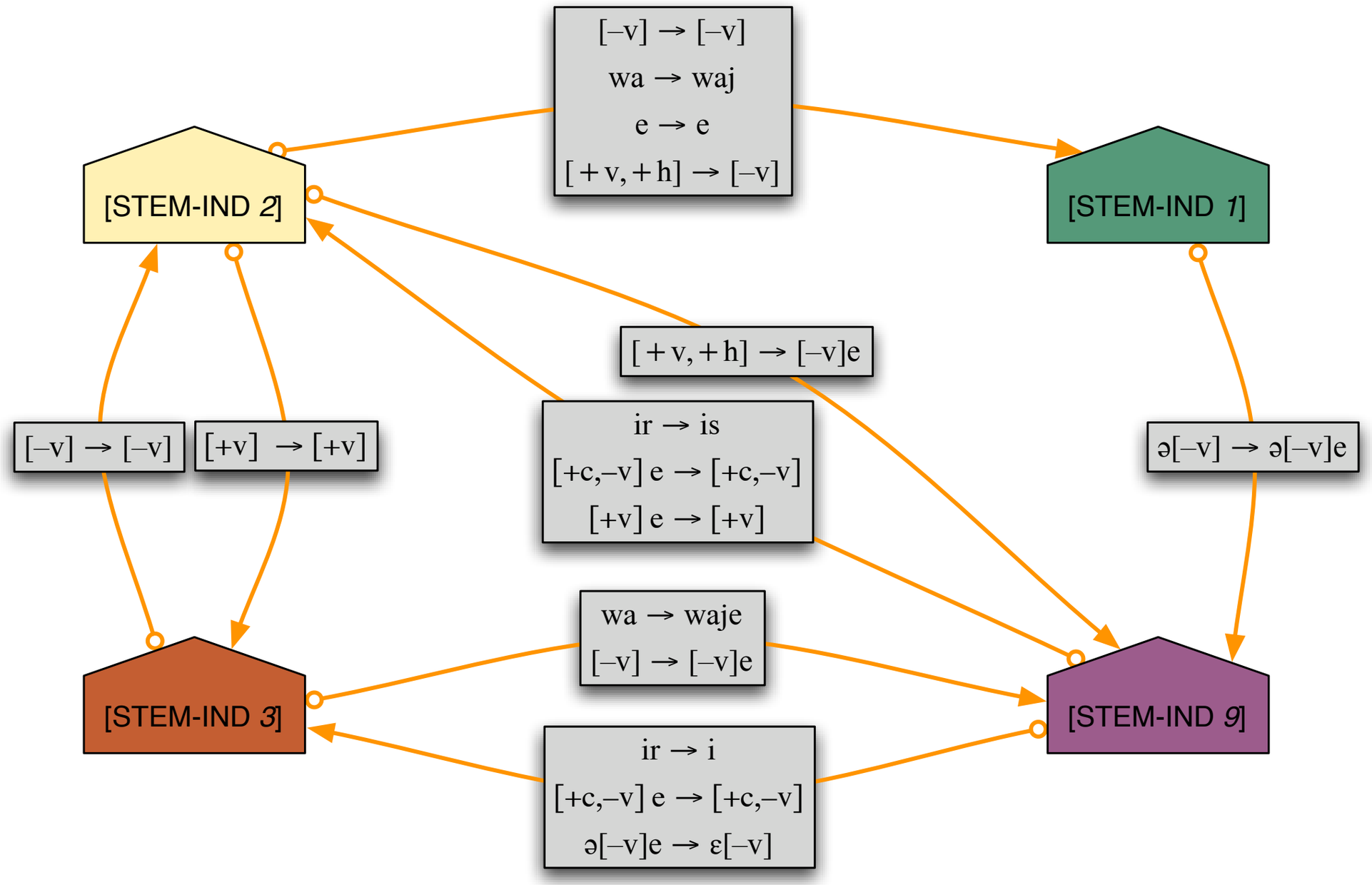
Problem 1: converse opacities

- Same reasoning holds for all other stems or forms, e.g. the infinitive



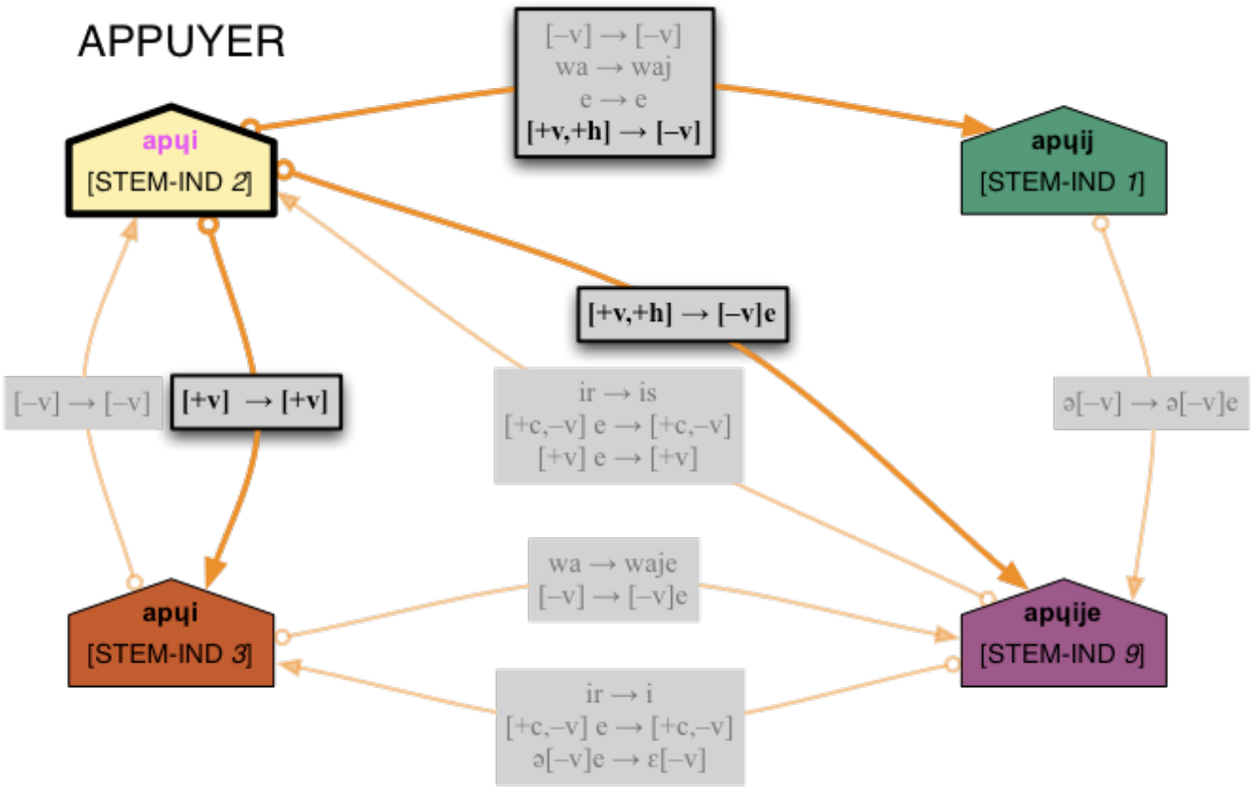
Solution 1: more connected, less predictive

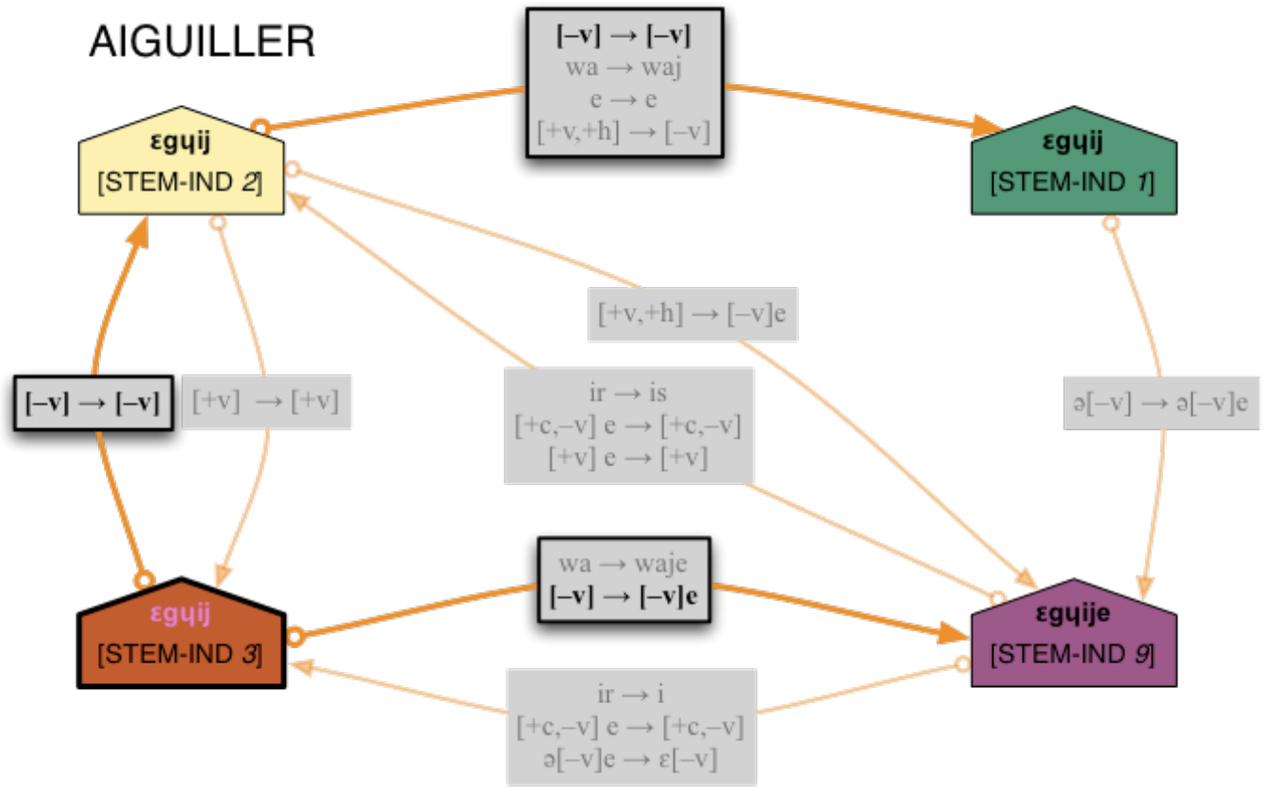
- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached



Solution 1: more connected, less predictive

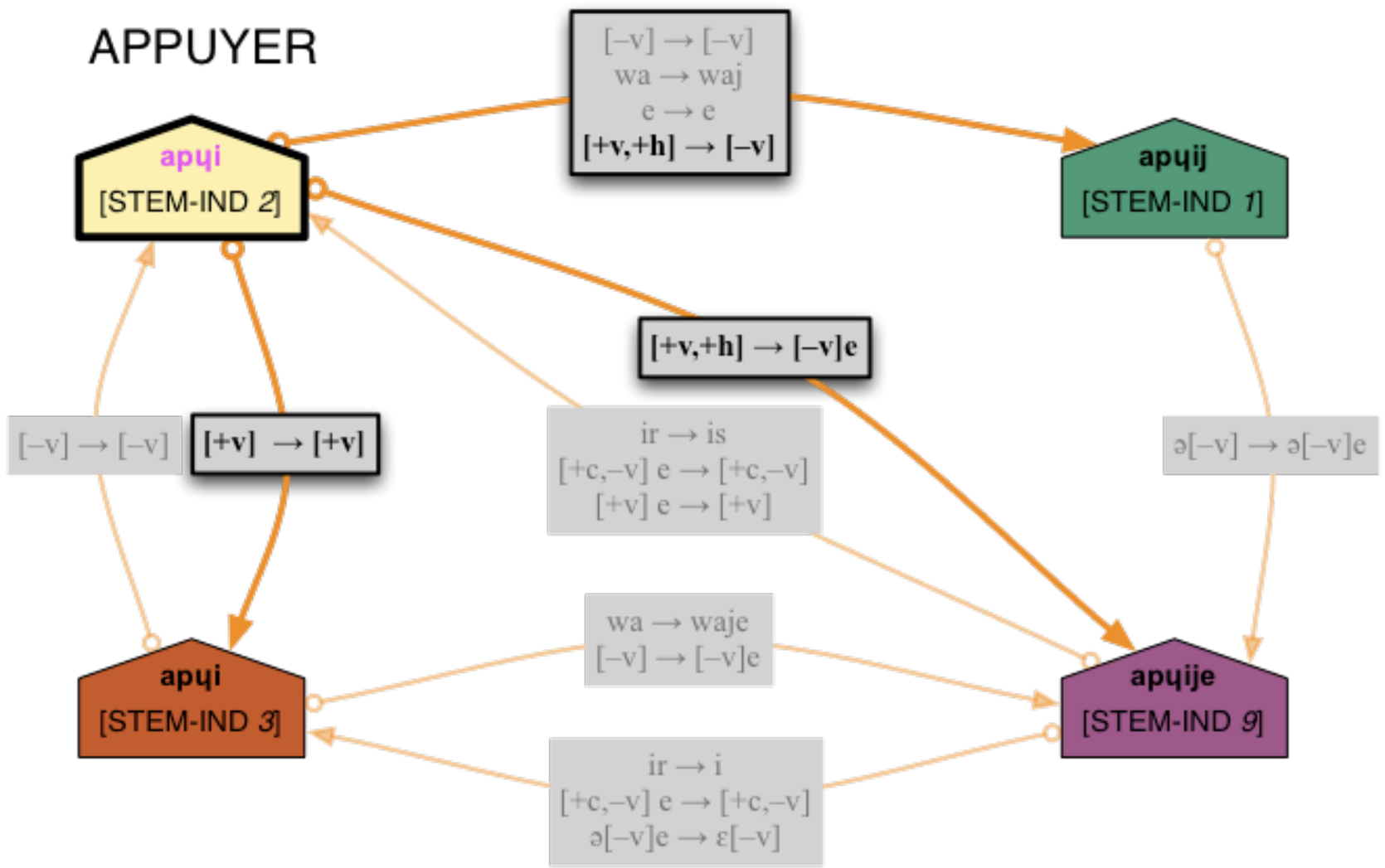
- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached





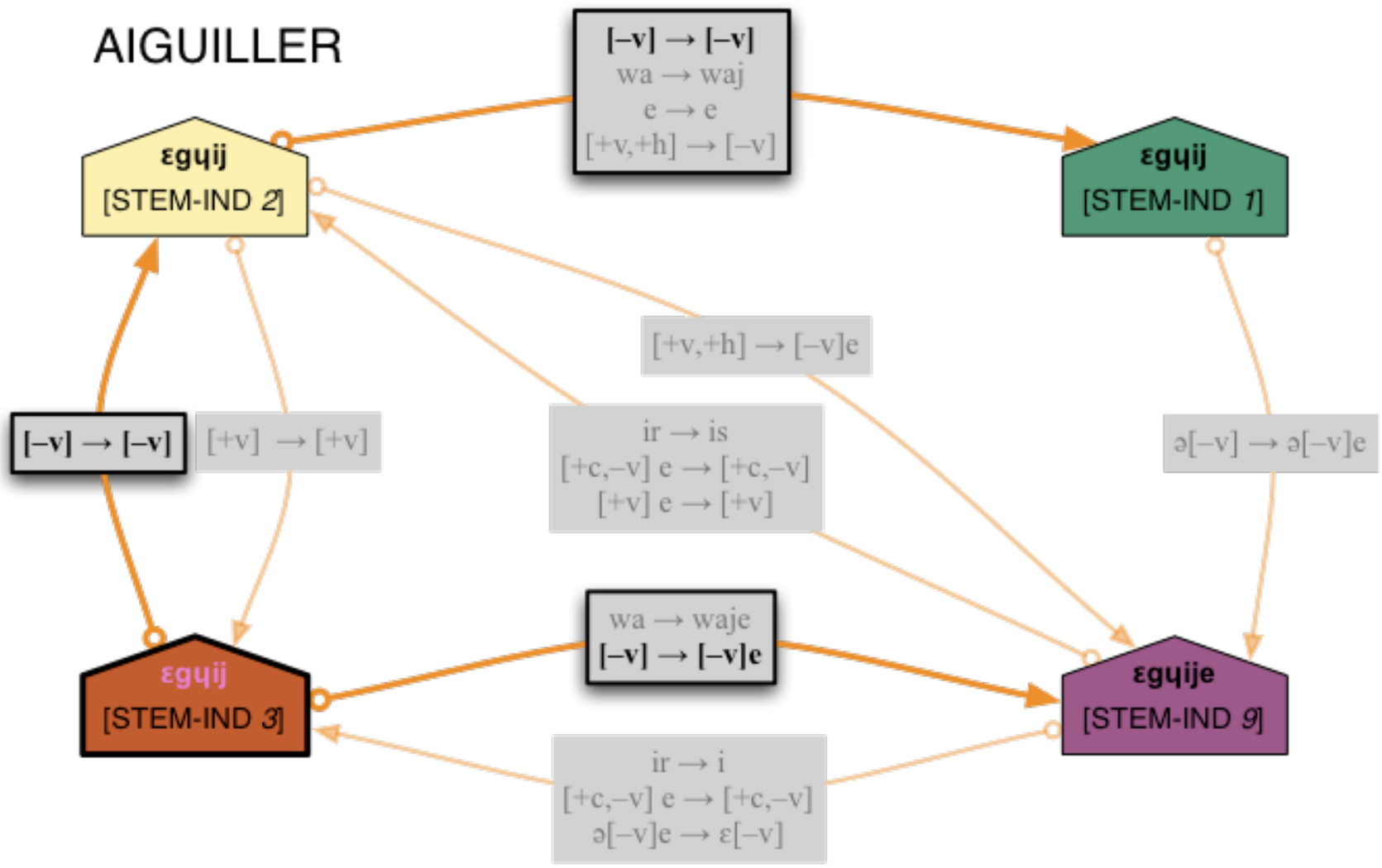
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached



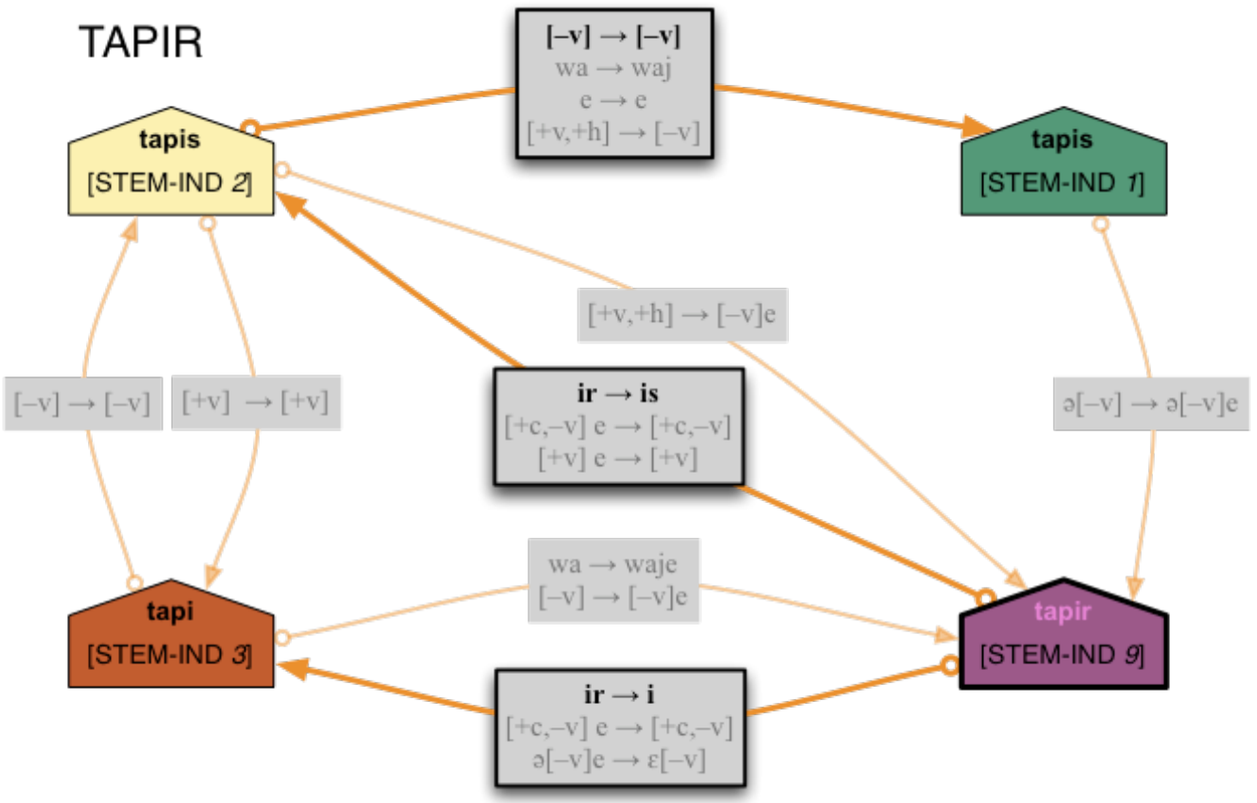
Solution 1: more connected, less predictive

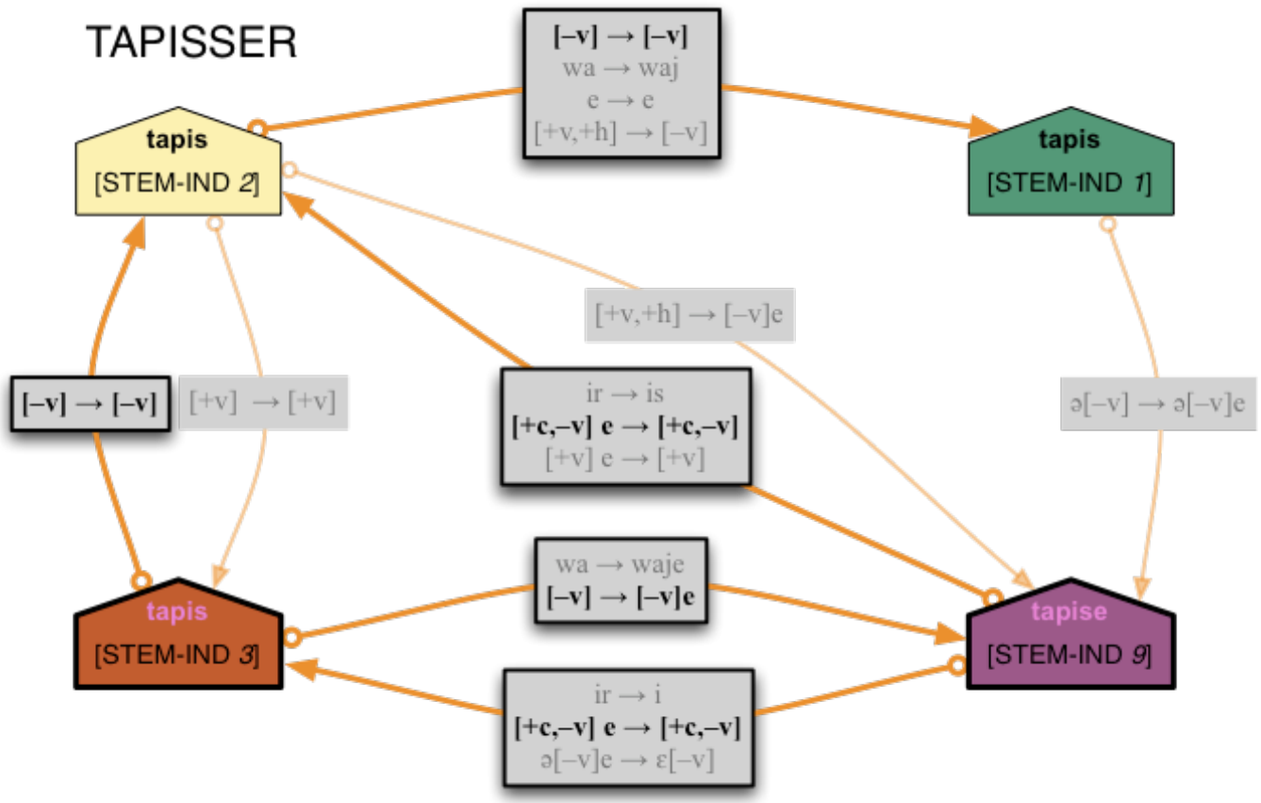
- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached



Solution 1: more connected, less predictive

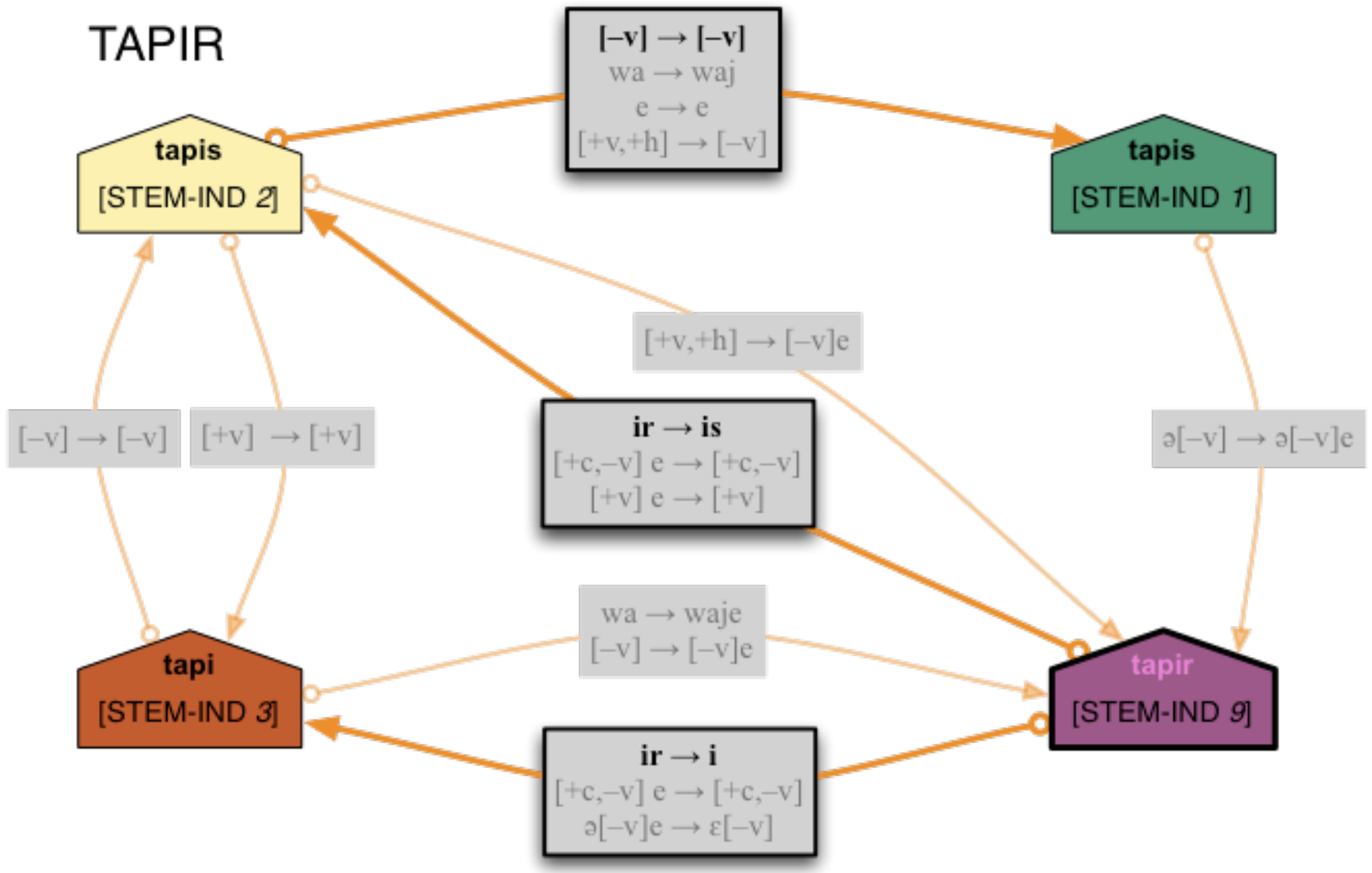
- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached





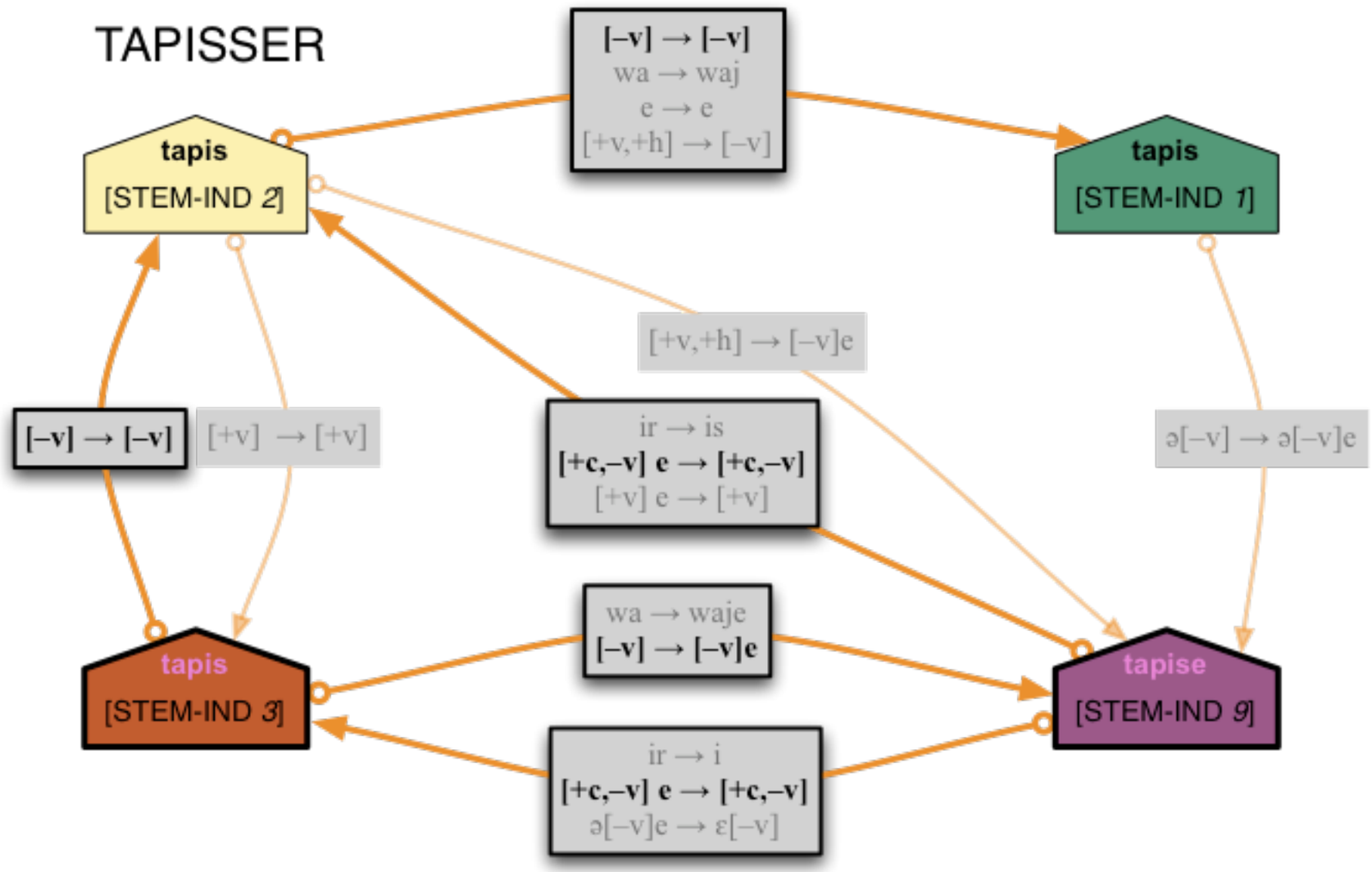
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached



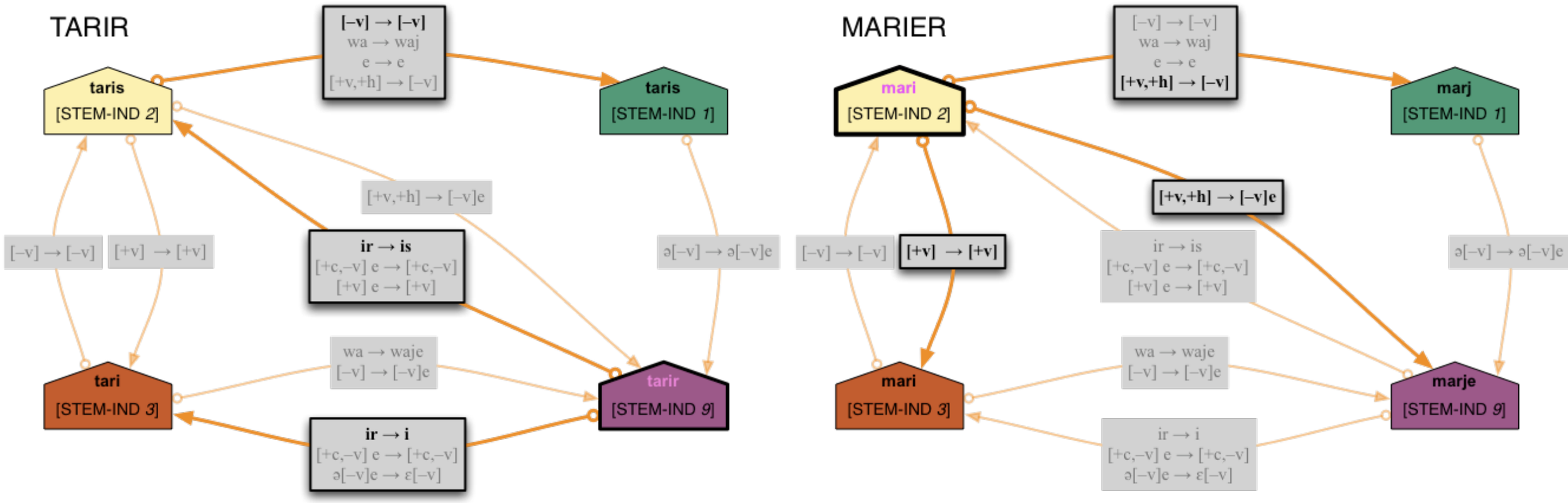
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached



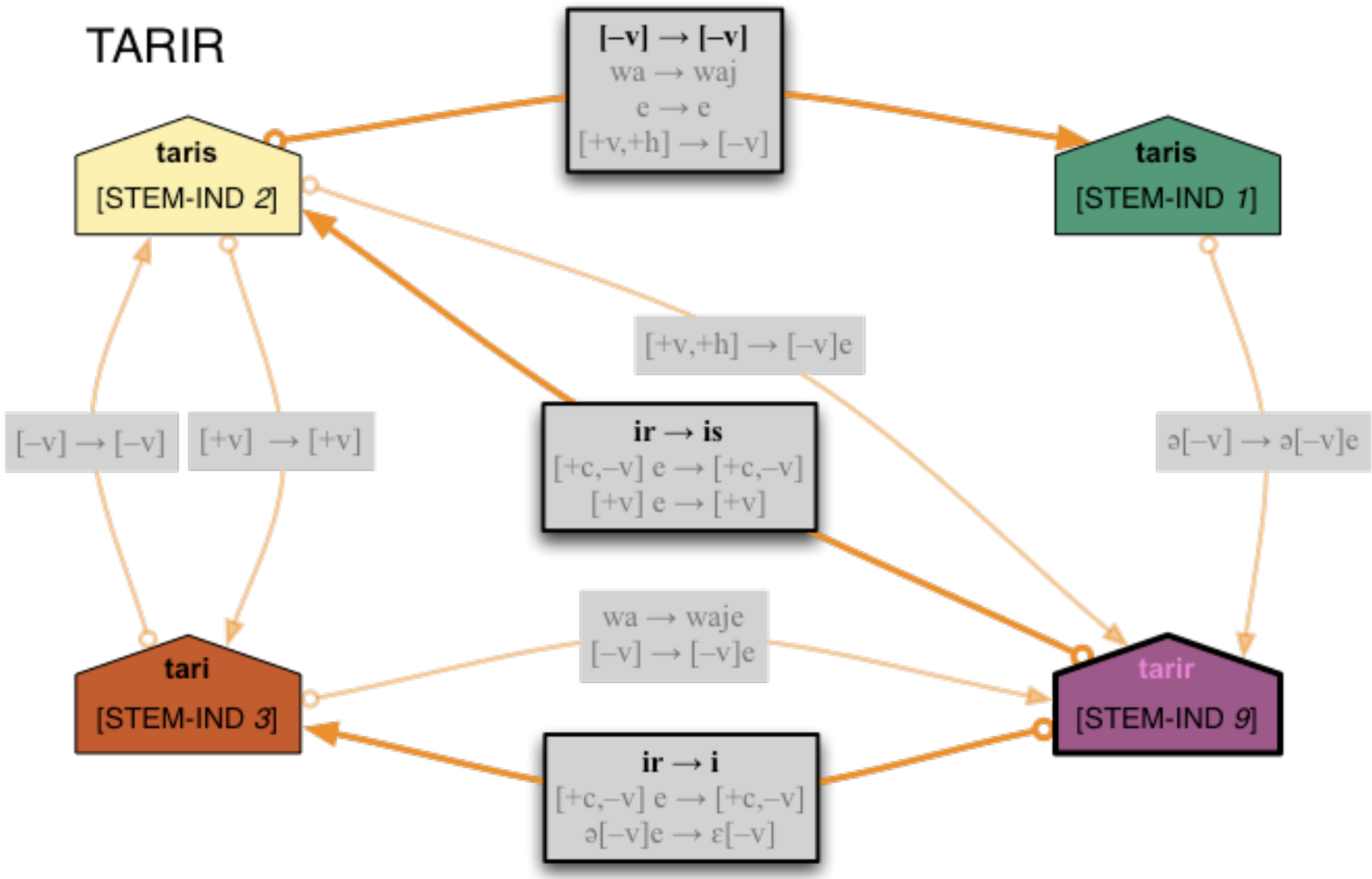
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached



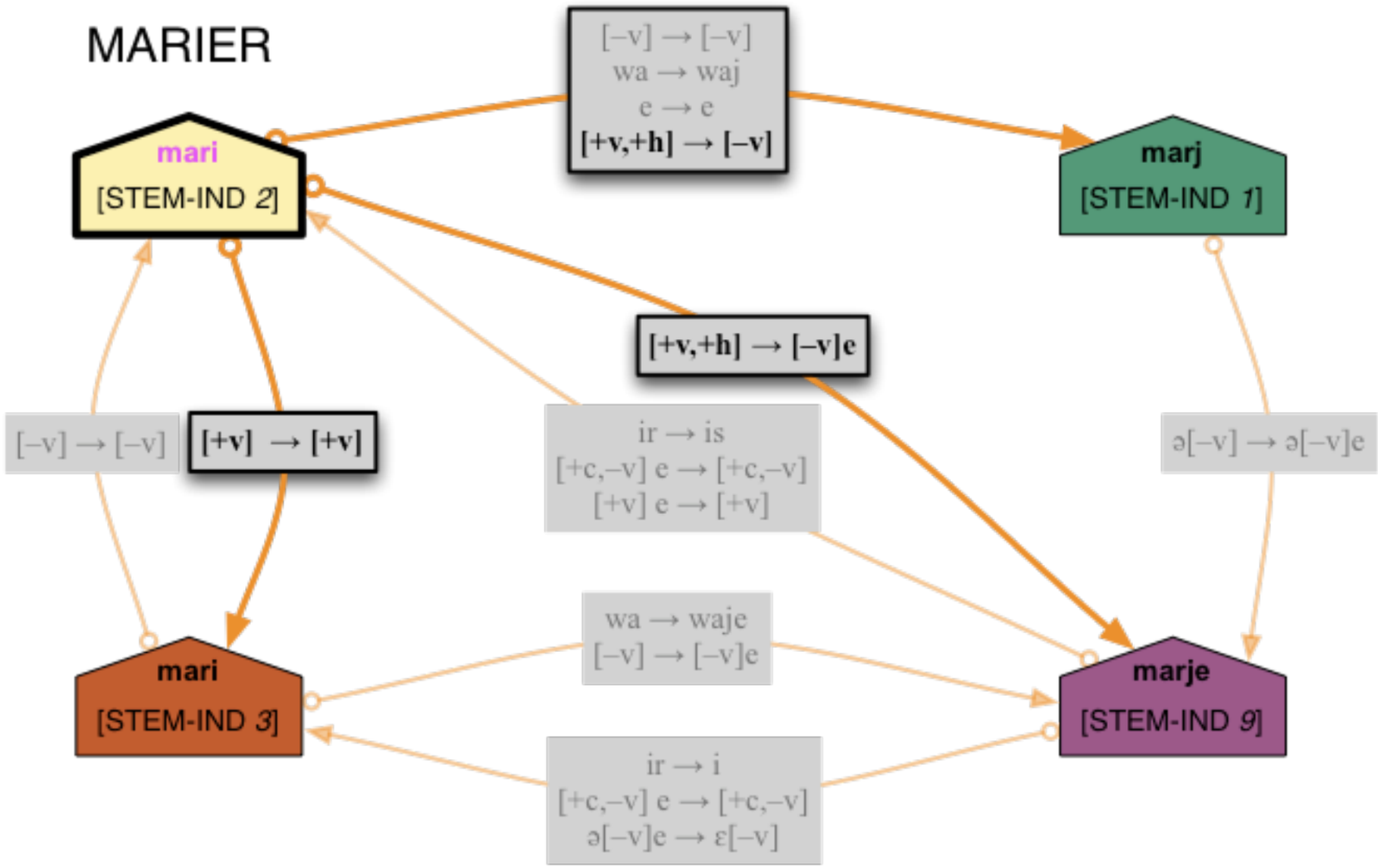
Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached



Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached



Solution 1: more connected, less predictive

- Change connections to partial functions
- Introduce more connections, so that for all regular verbs, there is at least one node from which all other nodes can be reached
- Note that:
  - There is still only one principal part for each regular French verb
  - Storing redundant information in the lexicon still works
  - Stating seperately the converse connections between two slots is a good idea after all

Problem 2: complex dependencies

- There are cases where a stem (resp. a form) can only be predicted on the basis of two other stems (resp. stems, forms)
  - A simple example: Masculine singular liaison forms of French adjectives (Bonami & Boye, 2005)

	MAS.SG	FEM.SG	MSLF
JOLI	ʒɔli	ʒɔli	ʒɔli
NET	nɛt	nɛt	nɛt
PETIT	pəti	pətit	pətit
COURT	kur	kurt	kurt
NOUVEAU	nuvo	nuvɛl	tapi

☞ The MSLF is identical to the FEM.SG if the MAS.SG is vocalic; else it is identical to the MAS.SG.

  - Other examples: languages with multiple principal parts for reguar lexemes

Problem 2: complex dependencies

- There are cases where a stem (resp. a form) can only be predicted on the basis of two other stems (resp. stems, forms)
  - In French conjugation: past participles (Bonami & Boye, 2006b)
    - For regular verbs participle formation is fully predictable from the infinitive stem: i → i, e → e
    - For irregulars, many attested patterns. But speakers have a strong preference for participles in y (see e.g. Kilani-Schoch & Dressler, 2005)

	INF	TRUE PP	COMMON ERROR
LAVER	lave	lave	—
FINIR	fini	fini	—
BOIRE	bwa	by	—
MORDRE	mɔrd	mɔrdy	—
PRENDRE	prɑ̃d	priz	prɑ̃dy
MOURIR	muri	mɔrt	mury
PEINDRE	pɛ̃d	pɛ̃t	pɛ̃dy

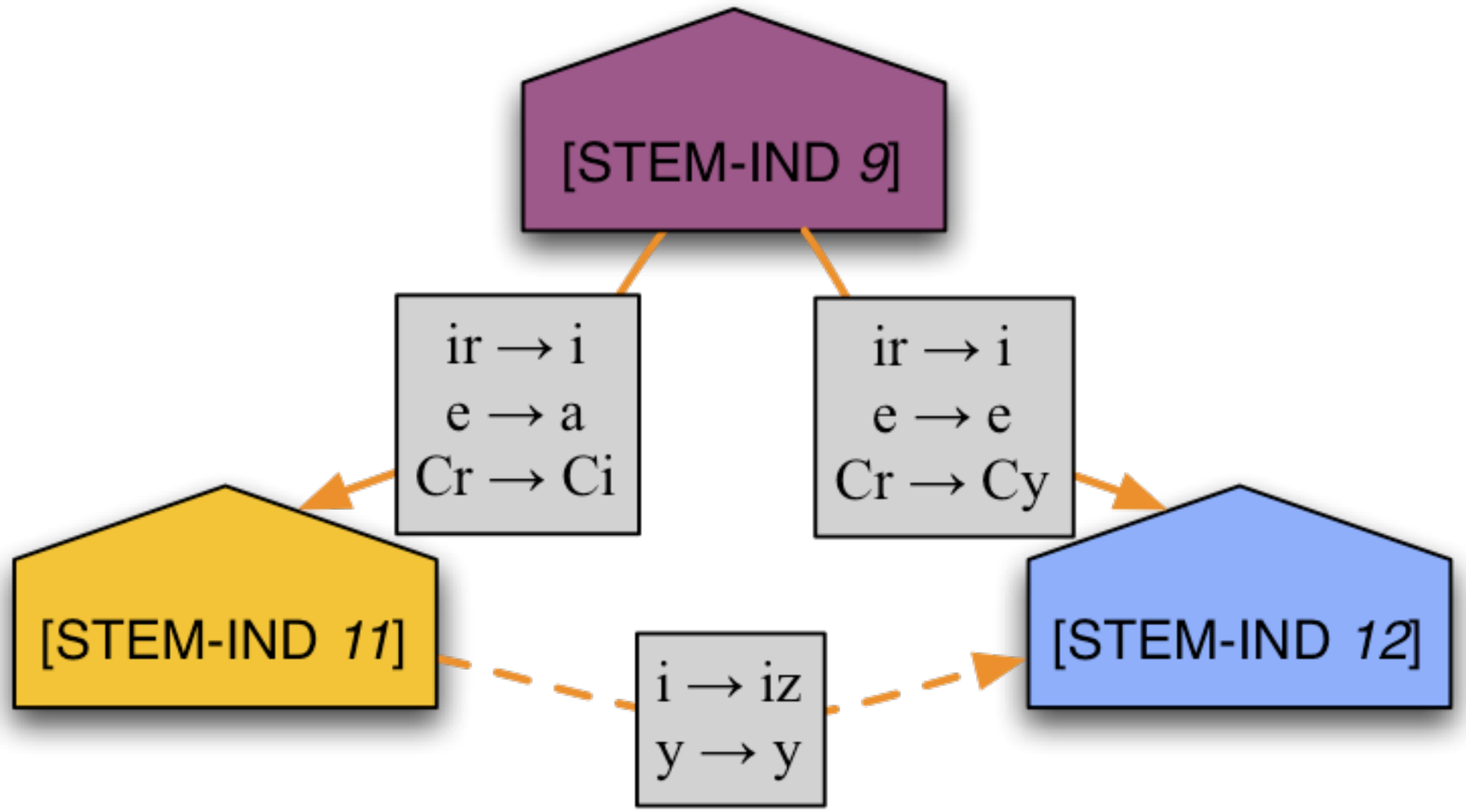
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:

	INF	SIMPLE PAST	TRUE PP	COMMON ERROR
LAVER	lave	lava	lave	—
FINIR	fini	fini	fini	—
BOIRE	bwa	by	by	—
MORDRE	mɔrd	mɔrdi	mɔrdy	—
PRENDRE	prɑ̃d	pri	priz	prɑ̃dy
MOURIR	muri	mury	mɔrt	mury
PEINDRE	pɛ̃d	pɛnji	pɛ̃t	pɛ̃dy

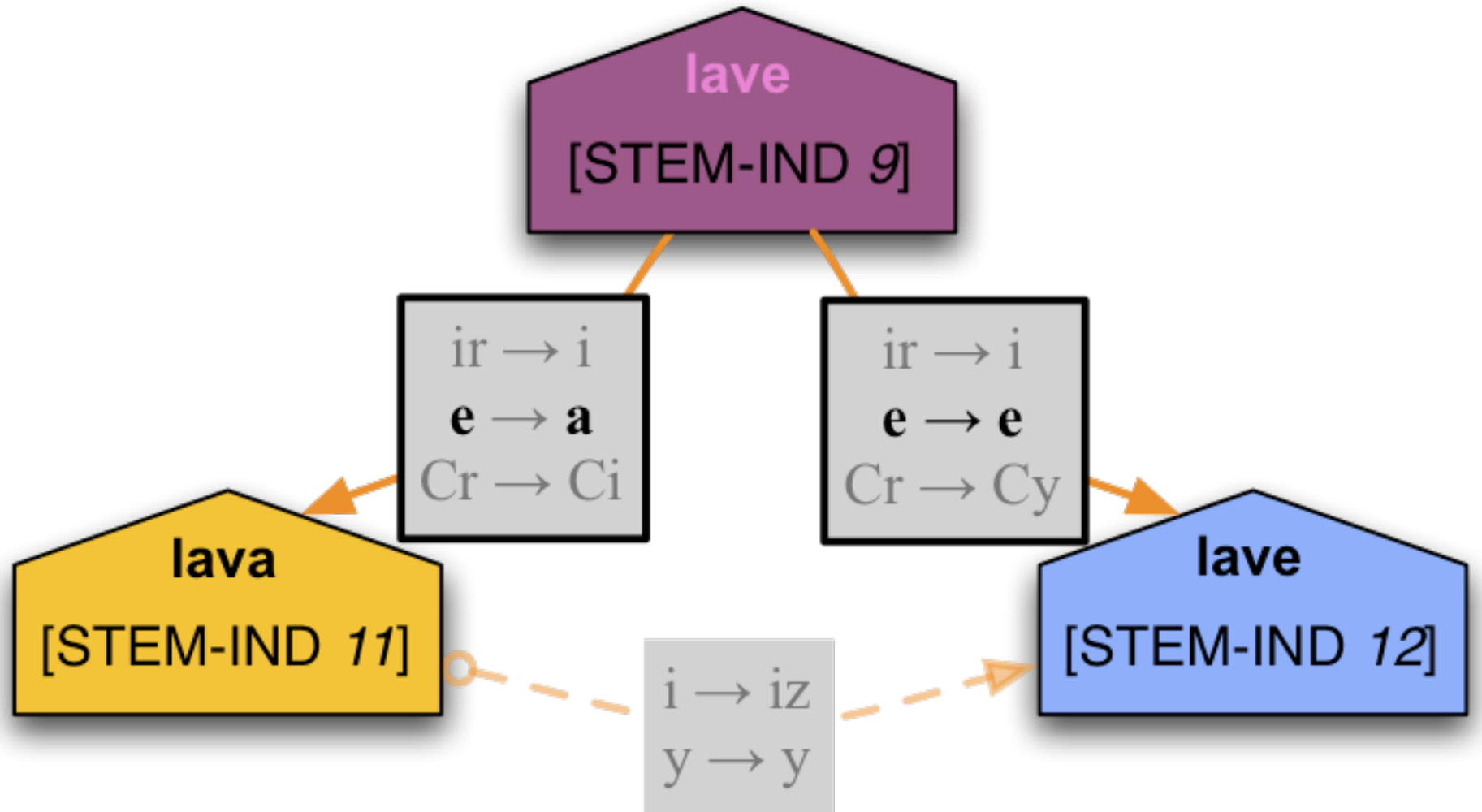
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:



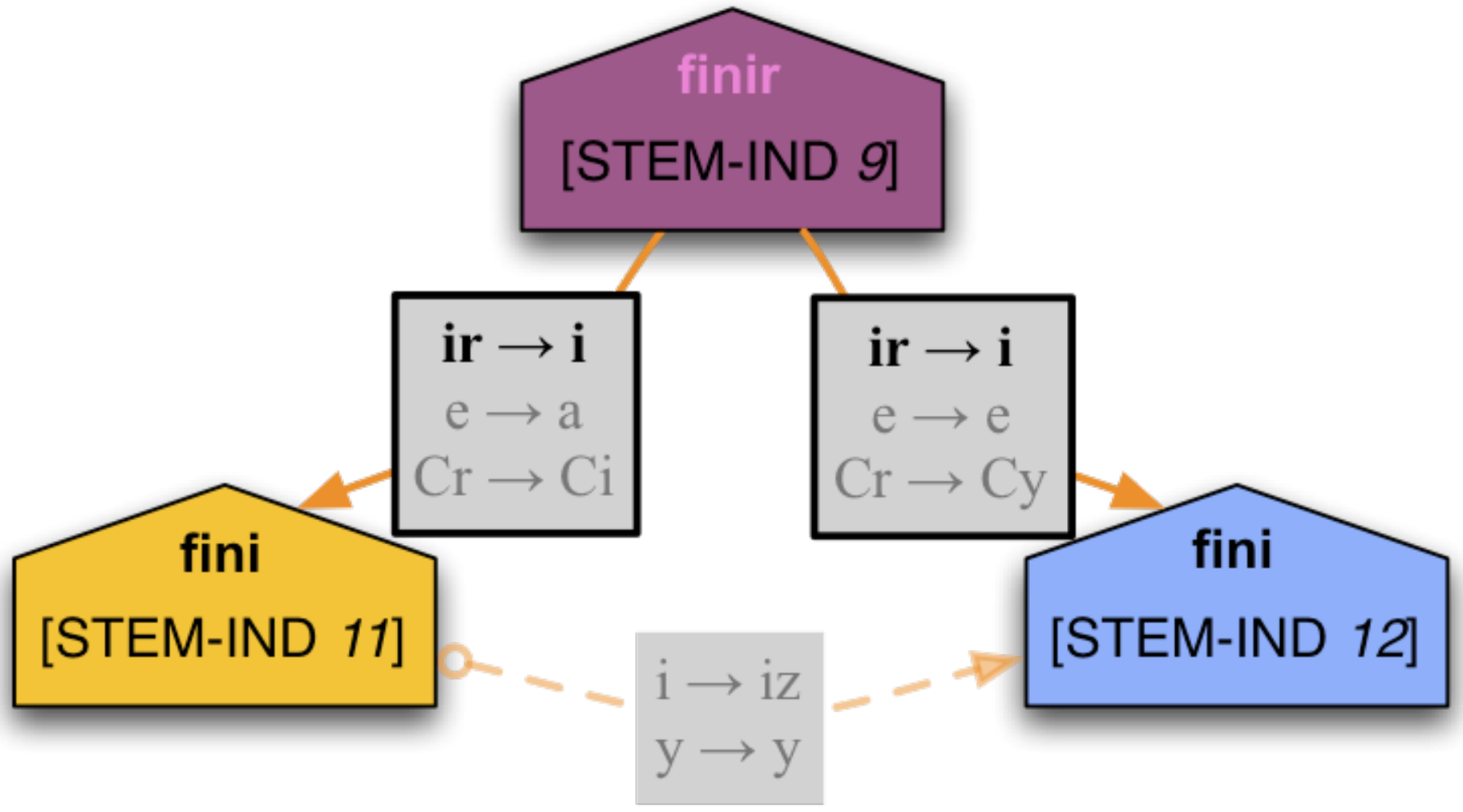
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - in the default case, predit both stem 11 and stem 12 from stem 9



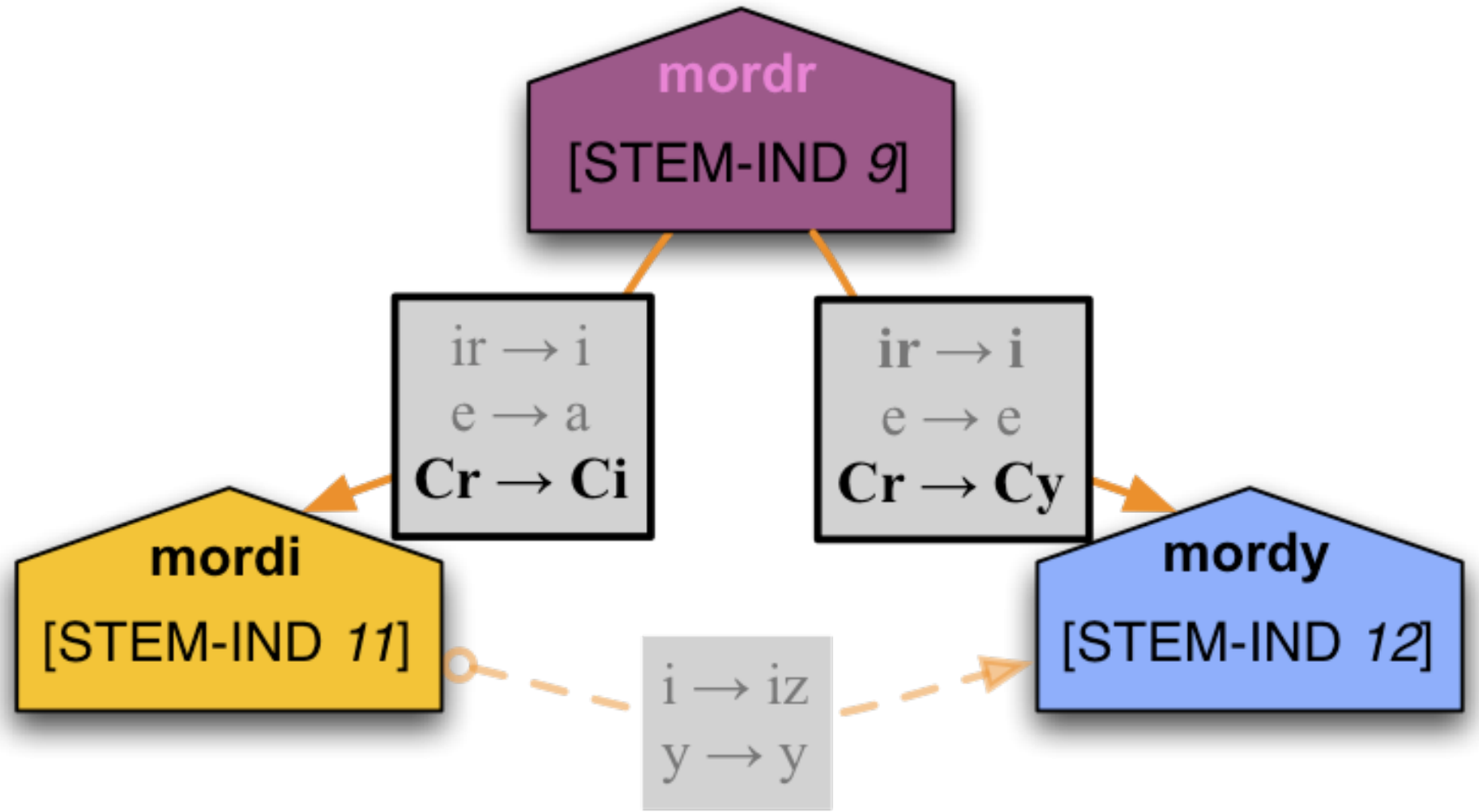
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - in the default case, predit both stem 11 and stem 12 from stem 9



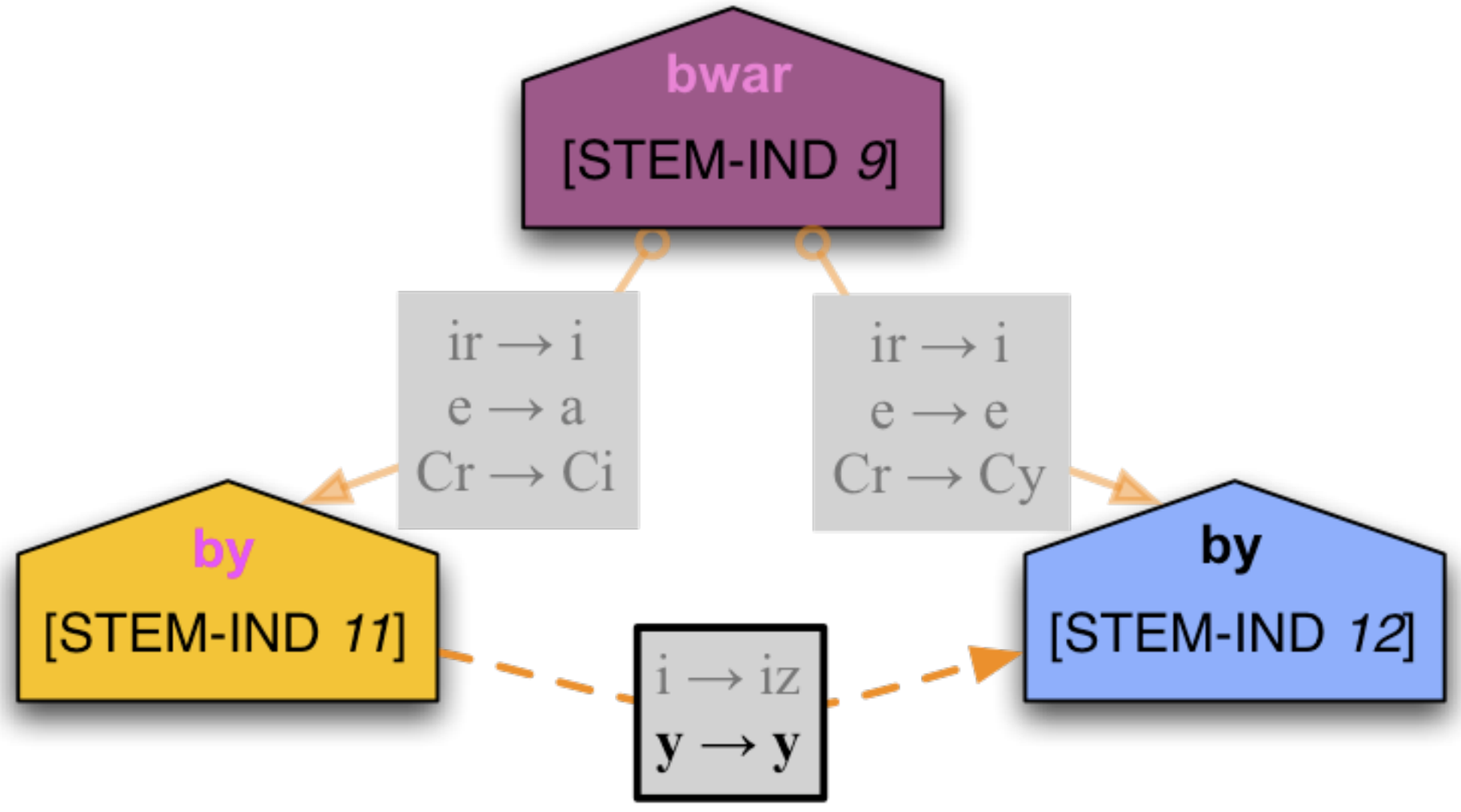
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - in the default case, predit both stem 11 and stem 12 from stem 9



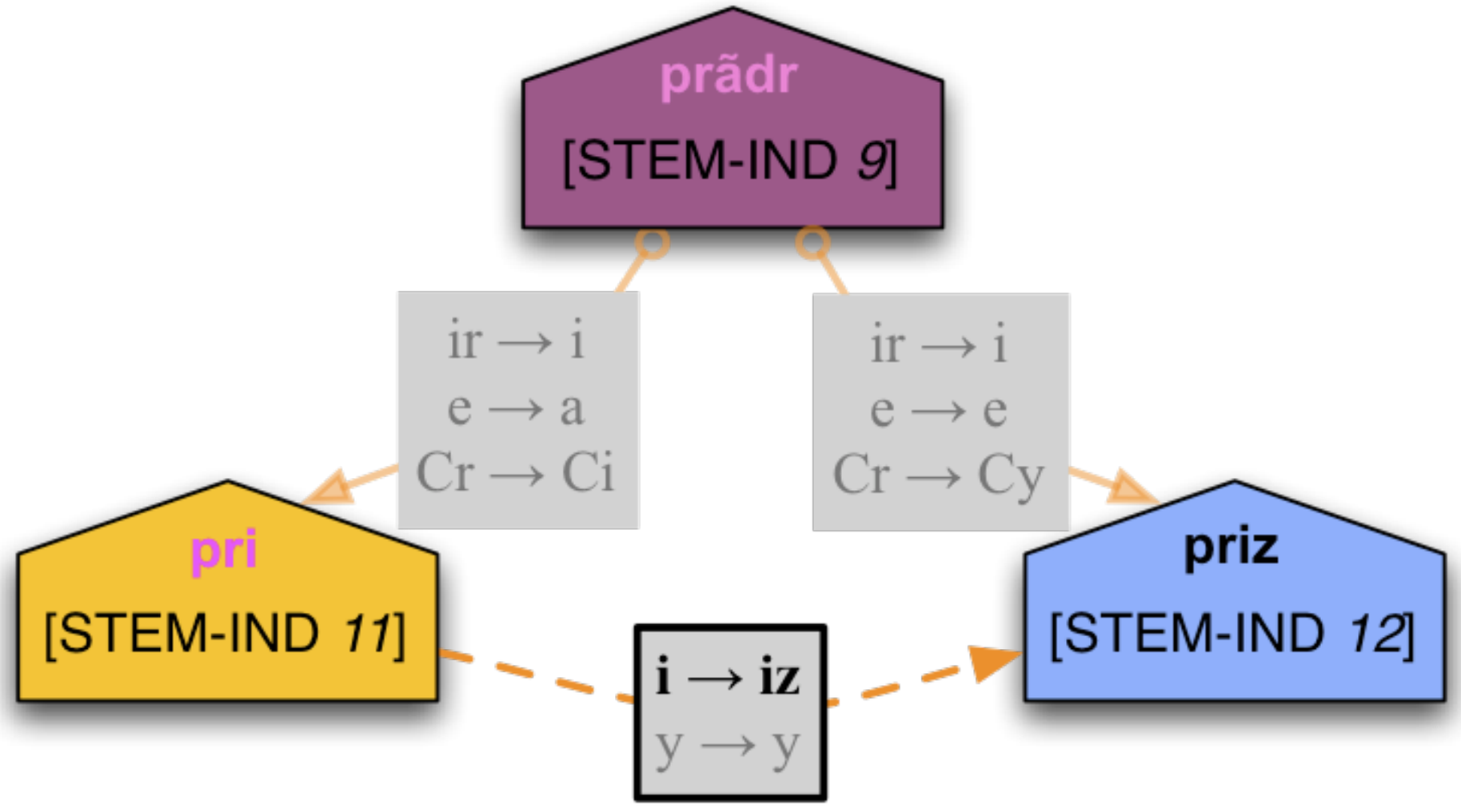
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - if stem 11 is not as predicted, then predict stem 12 from stem 11
  - if stem 12 is not as predicted, then store it



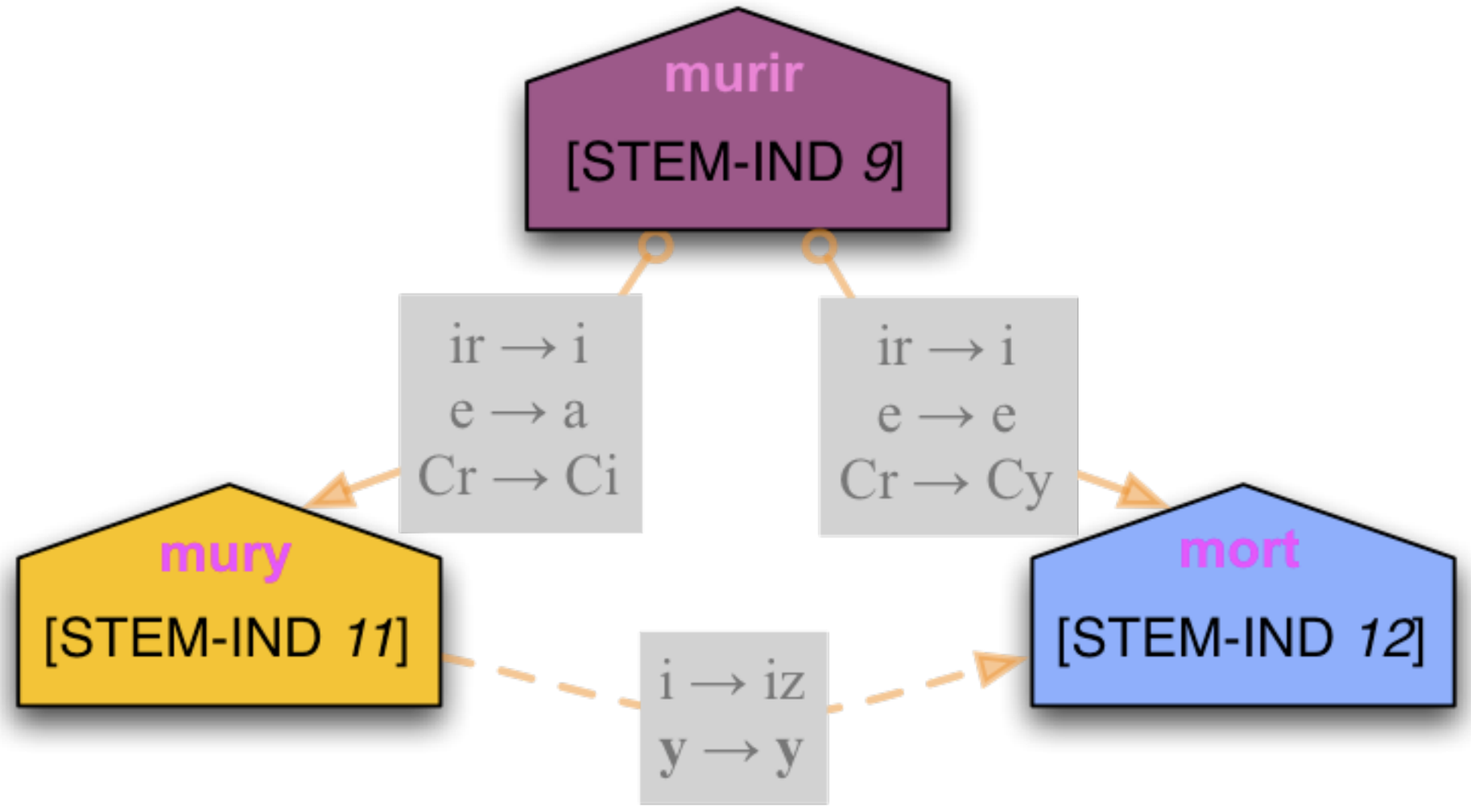
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - if stem 11 is not as predicted, then predict stem 12 from stem 11
  - if stem 12 is not as predicted, then store it



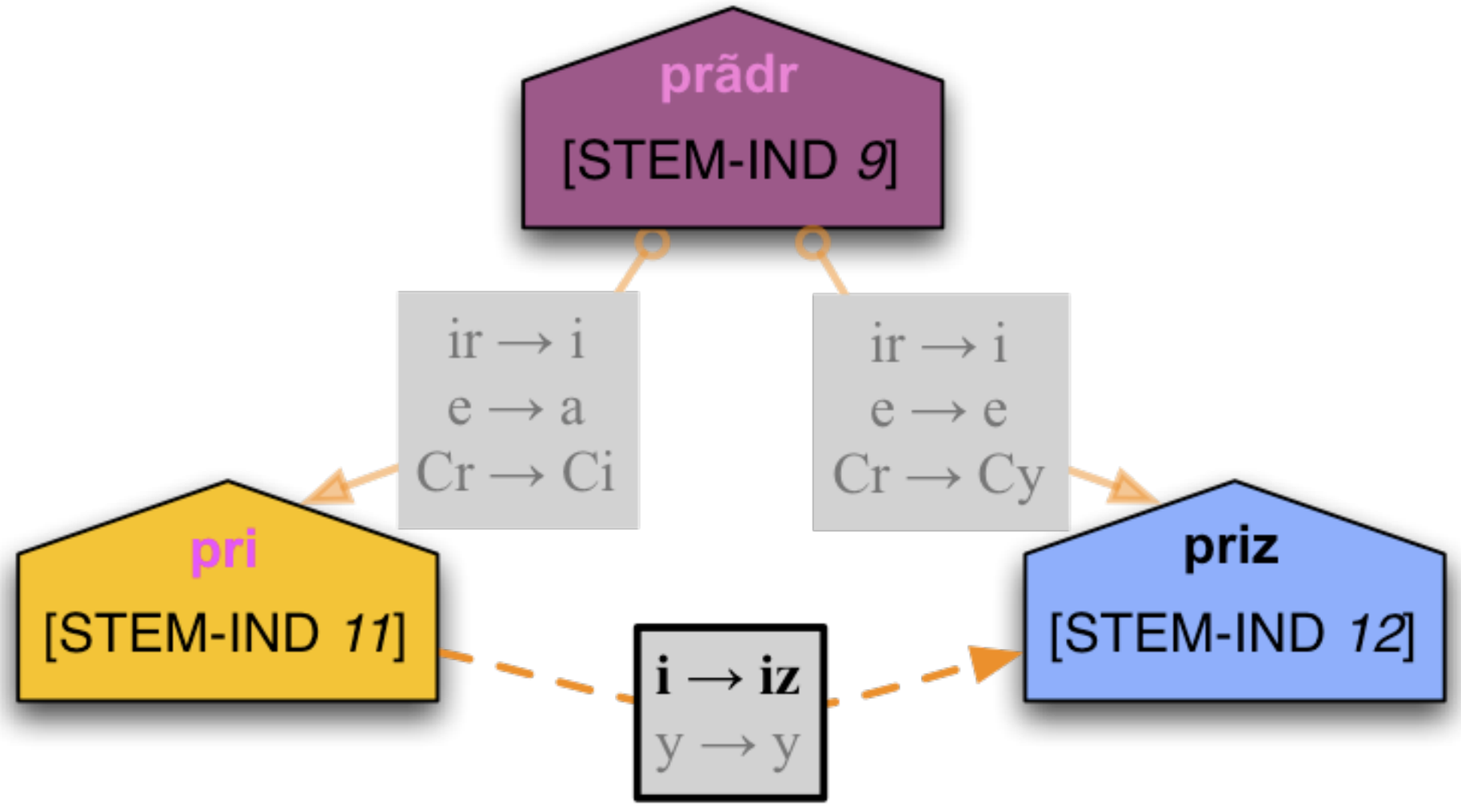
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - if stem 11 is not as predicted, then predict stem 12 from stem 11
  - if stem 12 is not as predicted, then store it



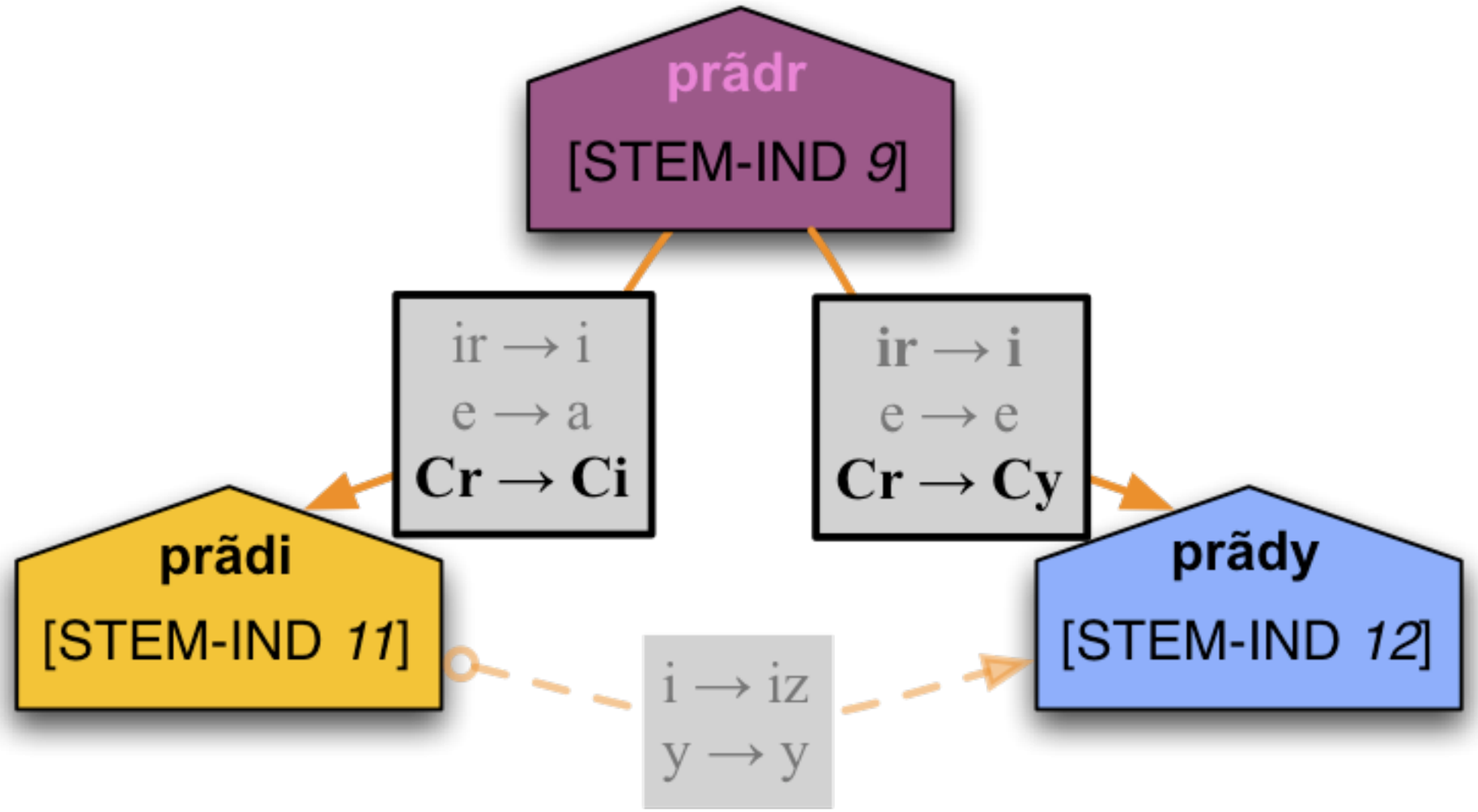
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - Errors arise when the speaker forgets about an unpredictable stem



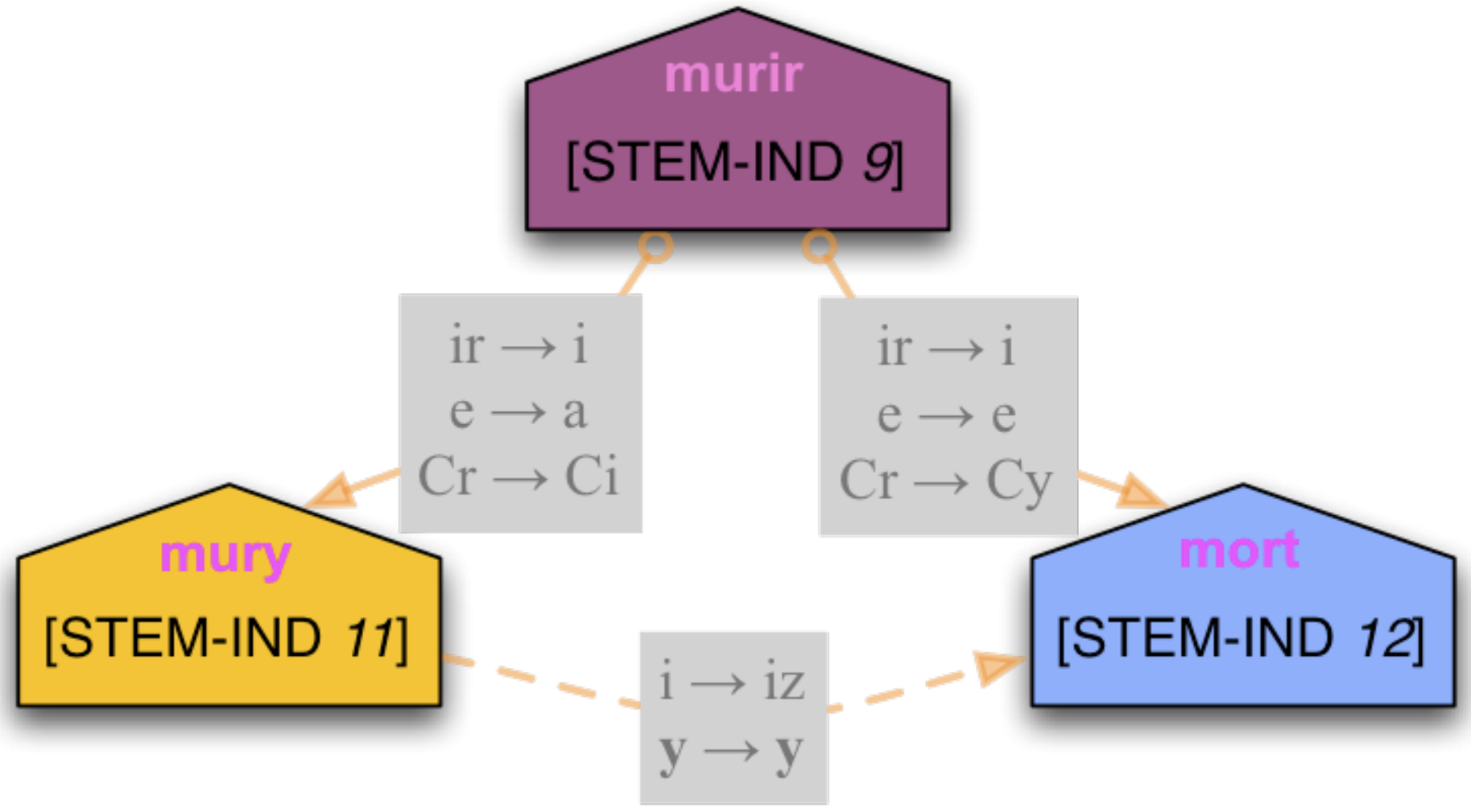
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - Errors arise when the speaker forgets about an unpredictable stem



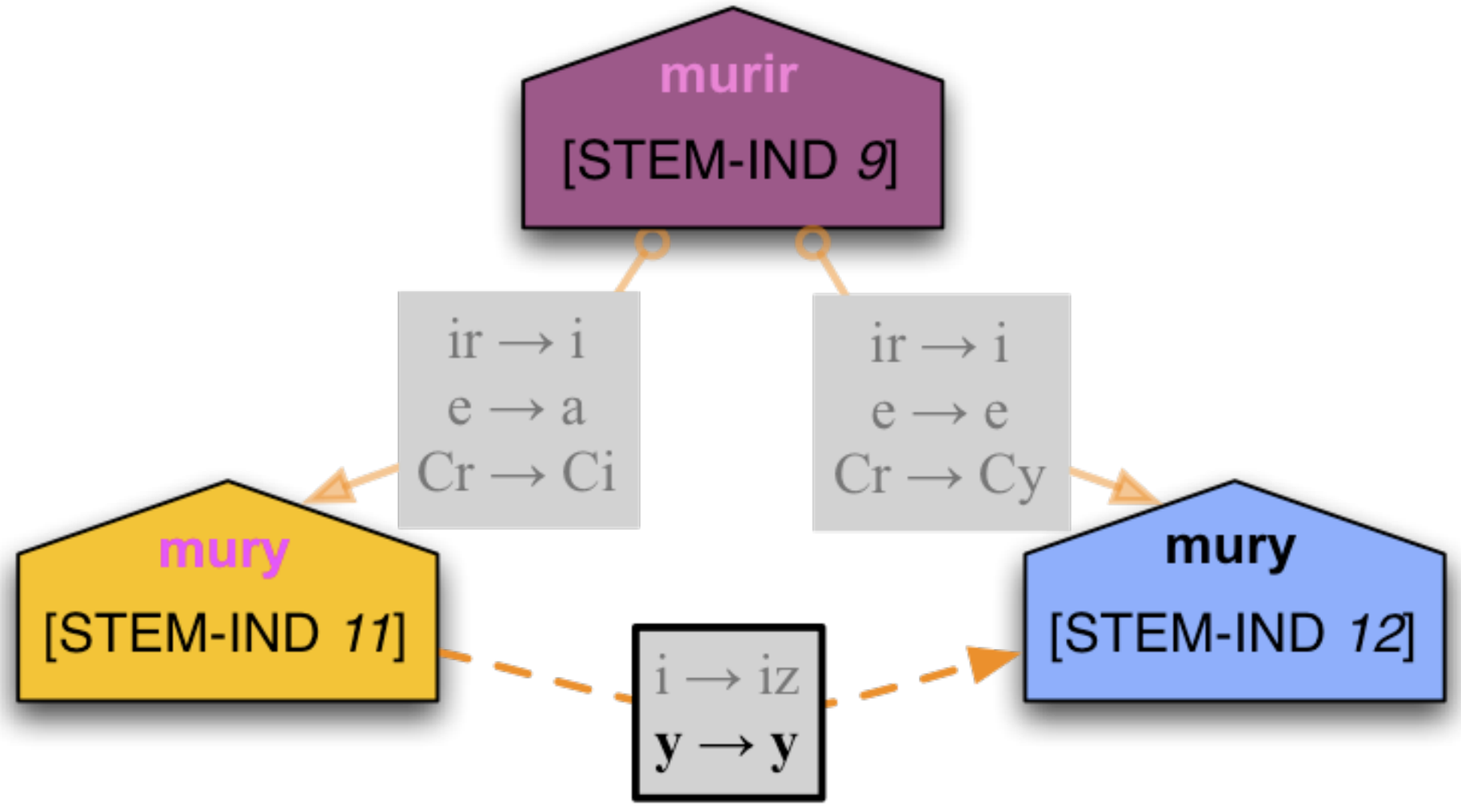
Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - Errors arise when the speaker forgets about an unpredictable stem



Problem 2: complex dependencies

- Taking the simple past into account allows better predictions:
  - Errors arise when the speaker forgets about an unpredictable stem



Solution 2: n-ary inflectional dependencies

- Up to now
  - The inflectional network is a graph connecting pairs of paradigm elements
  - The arcs in the graph are modelled by feature structure desctiptions with a single DTRS element.
- To account for complex dependencies we generalize
  - From pairs of paradigm elements to tuples of paradigm elements
  - Thus, to a network that is **not** a graph (not defined by a binary relation)
  - Formally, complex dependencies are modelled by feature structure descriptions with more than one DTRS element

Conclusions

Summing up

- Starting from the view of inflection as a network of forms:
  - We argued that stem spaces are motivated in French to account for suppletion patterns
  - We showed that stem spaces are fully compatible with an abstractive view of inflection
  - We discussed the benefits of fine-tuning the connectedness of the network of stems and forms
- At first sight this goes against Blevins's (2006) conclusions that:
  - Stems and roots are usually useless for the description of inflection patterns
  - In many cases starting from stems does not work, except through the postulation of nonobservables (class features, empty morphs, abstract phonology)

Summing up

- While we agree with the diagnosis, we think the illness comes from the search for minimal meaningful units, rather than abstract inflectional objects.
- Take French infinitives and present singulars:

	INF	PRST.SG	STEM 9
LAVER	lave	lav	lave
FINIR	finir	fini	fini
BOIRE	bwar	bwa	bwa
MORDRE	mərdr	mər	mərd
SORTIR	sərtir	sər	sərt

- Traditional approaches try to abstract away a minimal stem, and then need class features to account for infinitive endings
- In the current approach the choice of the stem is determined by suppletion patterns rather than minimality conditions
- Thus the stem is informative enough to serve as a principal part

An empirical test

- Thus the stem space is fundamentally a hypothesis on the relationship between stem suppletion and exponence
- What would be problematic would be a language where:
  - suppletion patterns motivate a stem space; but
  - prediction relations can not be seen on the stems

MAS.SG	MAS.PL	FEM.SG	FEM.PL
bak	bata	batu	baʃ
lup	luti	lutu	luʃ
mik	mita	mite	mis
sup	suti	sute	sus
sik	keta	kete	sis
uk	ama	amu	uʃ
pap	tisi	tisu	paʃ

An empirical test

- Thus the stem space is fundamentally a hypothesis on the relationship between stem suppletion and exponence
- What would be problematic would be a language where:
  - suppletion patterns motivate a stem space; but
  - prediction relations can not be seen on the stems

MAS.SG	MAS.PL	FEM.SG	FEM.PL	STEM 1	STEM 2
bak	bata	batu	baʃ	ba	bat
lup	luti	lutu	luʃ	lu	lut
mik	mita	mite	mis	mi	mit
sup	suti	sute	sus	su	sut
sik	keta	kete	sis	si	ket
uk	ama	amu	uʃ	u	am
pap	tisi	tisu	paʃ	pa	tis

An empirical test

- Thus the stem space is fundamentally a hypothesis on the relationship between stem suppletion and exponence
- What would be problematic would be a language where:
  - suppletion patterns motivate a stem space; but
  - prediction relations can not be seen on the stems