Rules and their ordering

Amba Kulkarni

Fellow IIAS, Shimla

2nd April 2017

Types of *sūtras*:

saṃjñā ca paribhāṣā ca vidhirniyama eva ca
atideshodhikāraśca ṣaḍvidham sūtralakṣaṇam

1. *saṃjñā sūtras*
   Rules which assign a technical term to a given entity
2. *paribhāṣā sūtras*
   Rules which regulate the proper interpretation of a rule
3. *vidhi sūtras*
   Rules which state a given ooperation to be performed on a given input under specific conditions
4. *niyama sūtras*
   Rules which restrict yje scope of a given rule
5. *atideśa sūtras*
   Rules which expand the scope of a given rule
6. *adhikāra sūtras*
   Rules which introduce a domain of rules sharing a common topic / operation etc.

Examples

- *saṃjñā sūtras*
  1.1.1 वृद्धिरादैच् (आ, ऐ and औ are called वृद्धि)
  1.1.2 अदेङ्गुणः (अ, ए and ओ are called गुण)
- *paribhāṣā sūtras*
  1.1.66 तस्मिन् इति निर्दिष्टे पूर्वस्य
  When a term is exhibited in the seventh case in these sūtras, the operation directed, is to be understood as effecting the state of what immediately precedes the one which the term denotes.
  1.1.67 तस्मात् इति उत्तरस्य
  An operation caused by the exhibition of a term in the Ablative (5th case), is to be understood to enjoin the substitution of something at the place of that which immediately follows the word denoted by the term.

Contd...

- *vidhi sūtras*
  6.1.77 इकोयणचि
  The semi-vowels य् व् र् ल् are the substitutes of the corresponding vowels इ उ ऋ and ऌ (long and short), when followed by a vowel.
- *adhikāra sūtras*
  6.1.72 संहितायाम् In the following sutras up to 6.1.157 inclusive, the words संहितायाम् 'in an unbroken flow of speech', should be supplied.

Contd...

Division *adhyāya* wise:

- The first *adhyāya*:- Introduction to the work
  - The definitions of the majority of the technical terms used in the work.
  - Most of the metarules and
  - Some operational rules.

  The chapter contains rules mainly related to
  1. the indicators (*it*) (1.3.2-1.3.9),
  2. assignment of *ātmanepada* and *parasmaipada* suffixes (1.3.12-1.3.93),
  3. *kāraka* definitions (1.4.23-1.4.55) and
  4. *nipātas* (1.4.56-1.4.97).
- The second *adhyāya*:- It contains four major divisions:
  1. Rules of compounding (2.1.3-2.2.38)
  2. Assignment of cases (2.3.1-2.1.73)
  3. Number and gender of compounds (2.4.1-2.4.31)
  4. *luk* elision (2.4.58-2.4.84)

Contd...

Division *adhyāya* wise:

- The third, fourth and fifth *adhyāyas*:-
  1. Give all the suffixes that are there in the language.
  2. The third *adhyāya* contains the suffixes that are added to a verbal root,
  3. the other two contain the suffixes that are added to a nominal stem.
- The sixth, seventh and eighth *adhyāyas*:-
  1. *saṃhitā adhikāra* (6.1.72-6.1.157)
  2. word accent (6.1.158-6.2.199)
  3. *aṅga adhikāra* (6.4.1-7.4.97)
  4. The *tripādī* (8.2.1-8.4.68)

Contd...

Subroutines: Marking *anubandha*s

- *upadeśe ac anunāsika it 1.3.2*
- *hal antyam 1.3.3*
- *na vibhaktau tusmā 1.3.4*
- *ādiḥ ñiṭudavaḥ 1.3.5*
- *ṣaḥ pratyayasya 1.3.6*
- *cutū 1.3.7*
- *laśaku ataddhite 1.3.8*
- *tasya lopaḥ 1.3.9*

Contd...

If we take into account the '*anuvrutti*', the rules may be rewritten as

upadeśe

- *ac anunāsika (=)it 1.3.2*
- *hal antyam 1.3.3*
  - *na vibhaktau tusmā (=it)1.3.4*
- *ādiḥ*
  - *ñiṭudavaḥ (=it) 1.3.5*
  - *pratyayasya*
    - *ṣaḥ (=it) 1.3.6*
    - *cutū (=it) 1.3.7*
    - *laśaku (=it) ataddhite 1.3.8*

The parallel between the *Pāṇini*'s *sūtra*s and the computer algortihm

```
if(the input is from UPADE'SA)
    Mark the ANUNASIKA AC as INDICATOR
  if(the last var.na is HAL)
     if(it is neither VIBAKTI nor TUSMA)
        Mark it as INDICATOR
  if(the INITIAL var.na is either ~ni or .tu or .du)
     Mark it as INDICATOR
  if(the INITIAL var.na is a PRATYAYA) {
     if (it is .sa)
          MARK it as INDICATOR
     if(it is from ca varga or ta varga)
          Mark it as INDICATOR
     if(it is NOT TADDHITA)
        if(it is either la or 'sa or ka varga)
          Mark it as INDICATOR
  }
```

Contd...

The operations in *Pāṇini*'s grammar:

- assigning a name
- substitution
- insertion
- deletion

Syntax of Aṣṭādhyāyī

Example 1

padam(word)::= *subantam*(nominal form)
| *tiṅantam*(verbal form)
(suptiṅantam padam 1.4.14)
;
subantam (nominal form)::= *prātipadikam* (nom stem)*sup*(nom suf
prātipadikam(nominal stem)::= *kṛt* (noun derived from a verb)
(*kṛttaddhitasamāsāśca* 1.2.46)
| *taddhita*(noun derived from a noun)
| *samāsa*(compound)
| underived_prātipadikam(nominal stem
;
samāsa(compound)::= *alaukika vigrahaḥ*(Intermediate ling ex
;
alaukika vigrahaḥ::= *prātipadikam sup prātipadikam sup*
(*saha supā* 2.1.4)
;

Syntax of Aṣṭādhyāyī

```
A ::=  B
     | C
     ;
B ::=  D E
D ::=  F
     | G
     | H
     | I
     ;
H ::=  J
     ;
J ::=  K '+' L M '+' N
     ;
```

Syntax of Aṣṭādhyāyī

Example 1 word ::= noun
| verb
;
noun ::= stem nominal_suffix
;
stem ::= derived_stem
| bare_stem
;
derived_stem ::= derived_stem1
| derived_stem2
| derived_stem3
;
derived_stem1 ::= verbal_root kṛt_suffix
;
derived_stem2 ::= nominal_root taddhita_suffix
;
derived_stem3 ::= nominal_root nominal_suffix nominal_root nominal_suffix

Example 2:
iko yaṇaci (6.1.77)

ik{6} yaṇ{1} ac{7}

Example 2:
iko yaṇaci (6.1.77)

ik{6} yaṇ{1} ac{7}

- tasmin iti nirdiṣṭe pūrvasya (1.1.65)
  A word ending in locative case indicates ‘of the preeceding’.
- *ṣaṣṭhī sthāneyogā* (1.1.48)
  A word in genitive case undergoes change.

Example 2:
iko yaṇaci (6.1.77)

ik{6} yaṇ{1} ac{7}

- tasmin iti nirdiṣṭe pūrvasya (1.1.65)
  A word ending in locative case indicates ‘of the preeceding’.
- *ṣaṣṭhī sthāneyogā* (1.1.48)
  A word in genitive case undergoes change.

ik{6} *ac*{7} ::= *yaṇ*{1} *ac*{7}

Example 2:
iko yaṇaci (6.1.77)

ik{6} yaṇ{1} ac{7}

- tasmin iti nirdiṣṭe pūrvasya (1.1.65)
  A word ending in locative case indicates 'of the preeceding'.
- *ṣaṣṭhī sthāneyogā* (1.1.48)
  A word in genitive case undergoes change.

ik{6} *ac*{7} ::= *yaṇ*{1} *ac*{7}
{i,u,ṛ,ḷ} {a,i,u,ṛ,ḷ,e,o,ai,au} → {y,v,r,l} {a,i,u,ṛ,ḷ,e,o,ai,au}

Syntax of Aṣṭādhyāyī

- tasmin iti nirdiṣṭe pūrvasya (1.1.65)
  A word ending in locative case indicates 'of the preceeding'.
- tasmāt iti uttarasya (1.1.66)
  A word ending in ablative case indicates 'of the following'.
- *ṣaṣṭhī sthāneyogā* (1.1.48)
  A word in genitive case undergoes change.

W{5} W{6}W{7} ::= W{5} W{1}W{7}

Ingermann observed that the sūtras of Aṣṭādhyāyī have the same structure as that of BNF and suggested to rename 'Backus Naur Form' as 'Pāṇini Backus Naur Form' (1967 ACM Communications).

Contd...

Ingenious usage of *vibhaktis*:
A typical context sensitive rule

$$\alpha\beta\gamma \rightarrow \alpha\delta\gamma$$

- 5^{th} case to indicate the left context
- 7^{th} case to indicate the right context
- 6^{th} case to indicate which element will undergo a change and
- 1^{st} to indicate what it will change to.

$$5\ 6\ 7 \Rightarrow 5\ 1\ 7$$

An example from *Aṣṭādhyāī*.

ataḥ roḥ aplutāt aplute (ut ati saṃhitāyām) 6.1.113
ataḥ{5} ru{6} aplut{5} apult{7} (ut{1} at{7})
apluta_at ru apluta_at ⇒ apluta_at ut apluta_at

ṡiva ru arcya ⇒ ṡiva u arcya

Another example:

iko yaṇ aci

ik ac → yaN ac

{ik} {ac} → {yaN} {ac}

{ik} = {i u ṛ ḷ }

{yaṇ} = {y v r l }

Rule 1.3.10 **यथासंख्यम् अनुदेशः समानम्** says 'When a rule involves the case of equal numbers of substitutes and of things for which these are to be substituted, their mutual correspondence or assignment of each to each, is according to the order of enumeration'.

Thus

i → y

u → v

ṛ → r

ḷ → l

Conflict Resolution

Programming Languages

- Longest match
- Precedence Rules

Aṣṭādhyāyī

- *vipratiṣedhe paraṃ kāryam* (1.4.2)
- *pūrvatra asiddham* (8.2.1)
- *asiddhavat atra ābhāt* (6.4.22)
- *ṣatvatukorasiddhaḥ* (6.1.83)

Contd...

How are the rules triggered?

Typical Grammarian's view:
The rules in the *sapāda saptādhyāyī* seek for an opportunity(*nimitta*) to act on an input. In case there is a conflict, there are certain conflict resolution techniques (described as *paribhāṣā*), which come into play.The *nimitta* is in the form of a context. When no other rules are applicable, then the rules in tripādī are applied sequentially thereby ending the process of derivation.

Contd...

This model is described as below.

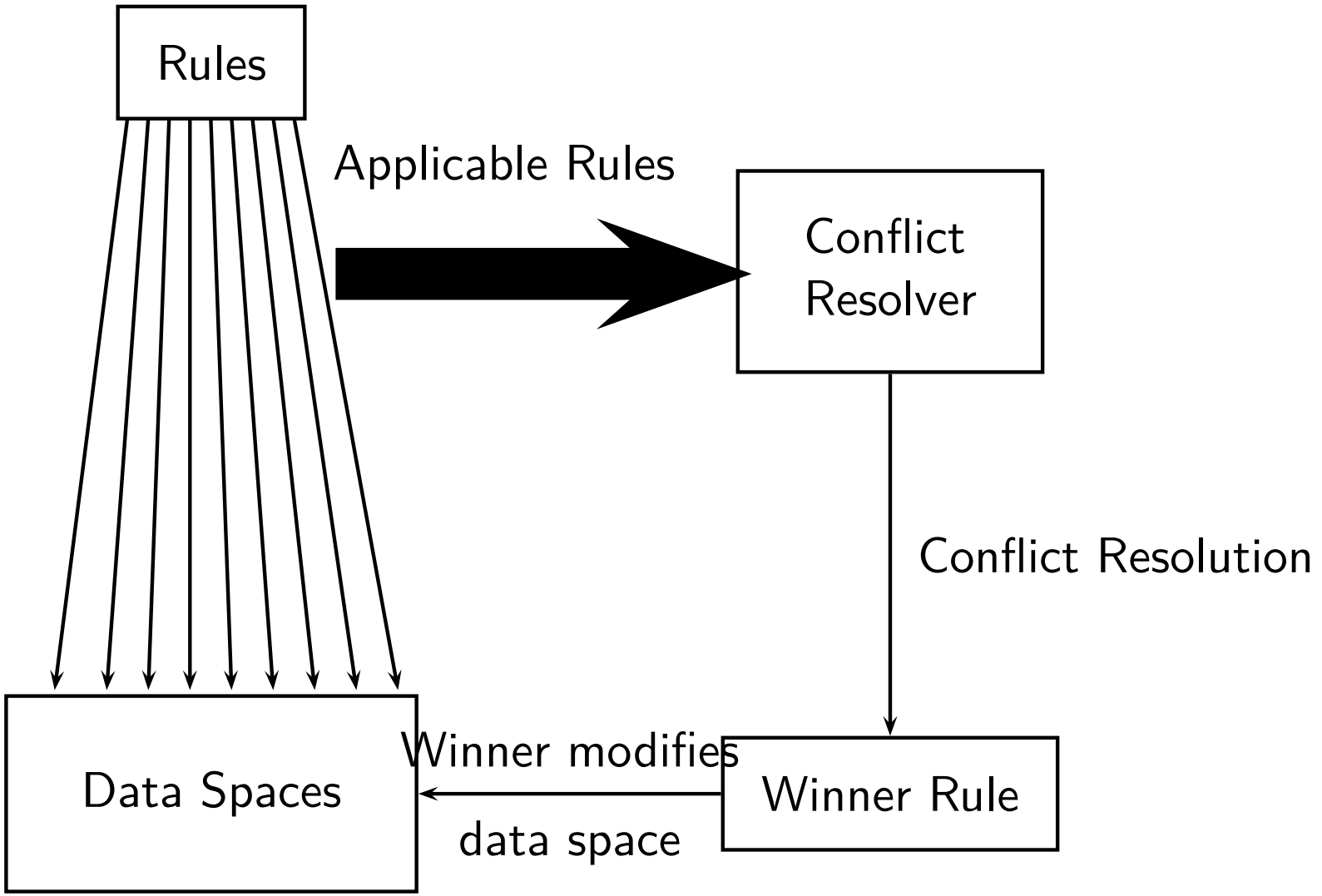


Figure : Application of rules

Contd...

Asiddhavat, Asiddha and Asiddham

- *pūrvatra* ***asiddham*** *8.2.1*
- ***asiddhavat*** *atra ābhāt 6.4.22*
- *ṣṭvatukor****asiddhaḥ*** *6.1.86*

Contd...

Asiddham- *pūrvatra* ***asiddham*** *8.2.1*

- The output of the rules in this part is not available to the earlier rules. $\Rightarrow$ *tripādī* should follow *sapāda saptādhyāyī*
- The rule makes the output of each of the following *sūtra* unavailable to the previous rules. $\Rightarrow$ The rules within *tripādī* should be followed lineraly.
- Whole *tripādi* may be considered to be a single subroutine

Asiddhavat- asiddhavat *atra ābhāt 6.4.22*

Consider the derivation: *śādhi* from *śās + hi*.

Applicable *sūtras*:

- *hujhalbhyo herdhi 6.4.101*
  After हु and after a consonant (with the exception of semi-vowels and nasals), धि is substituted for the imperative affix हि when the latter does not take the augment इट्
- *śā hau 6.4.35*
  Before the Imperative affix ह , शा is substituted for शास्

6.4.101: śās + hi → śās + dhi

6.4.35: śās + hi → śā + hi

Contd...

- Task parallelism is achieved.
  - If 6.4.101 is applied, then the conditions for applying 6.4.35 are not met.
  - If 6.4.35 is applied first, then the conditions for 6.4.101 would not be met.
  - *sàās + hi* $\rightarrow$ *sàā + hi*: Result invisible to 6.4.35.
- Economy is achieved. R: a b $\rightarrow$ c d factored as:

$$R_1 : ab \rightarrow cb, R_2 : ab \rightarrow ad$$

Asiddhaḥ- *ṣtvatukor***asiddhaḥ** *6.1.86*
'ekādeśa' section is invisible to the 'ṣatva' and 'tuk' processes.

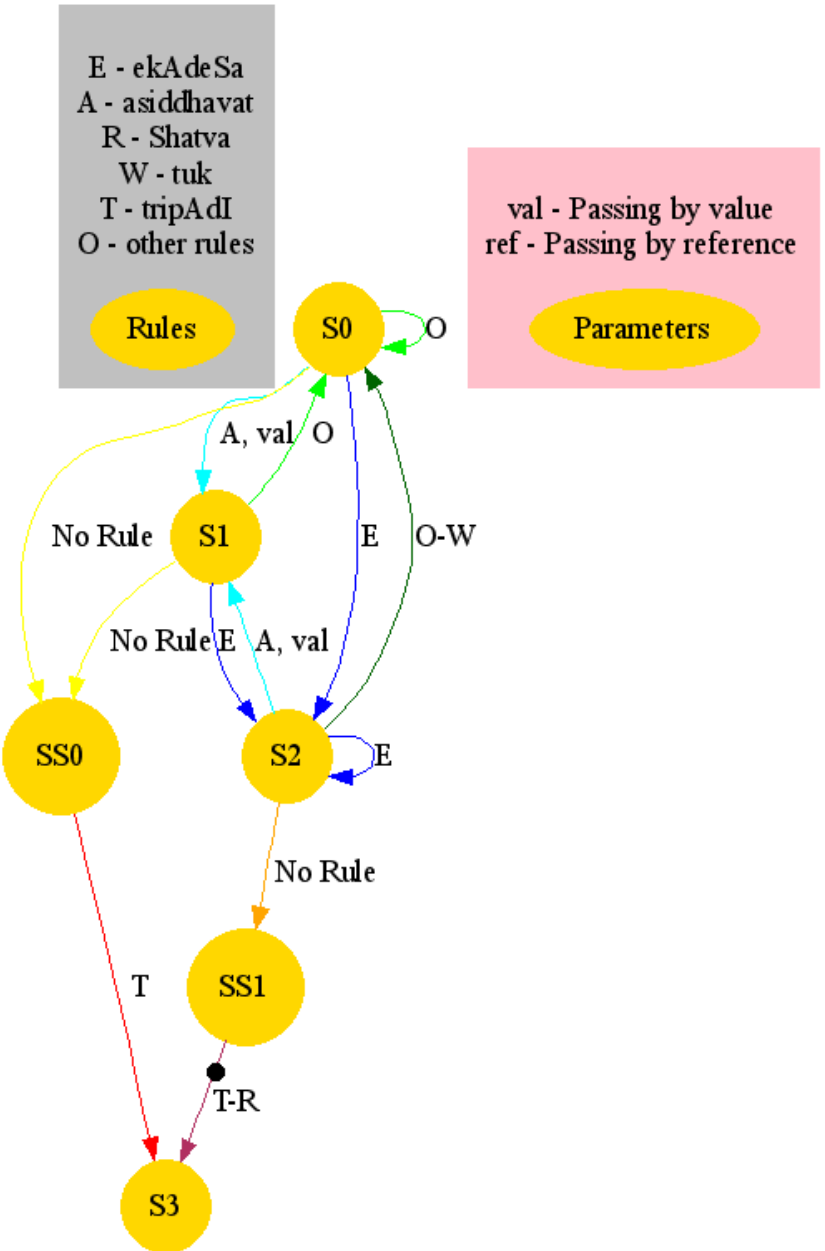


Figure : Model of *Aṣṭādhyāyī*

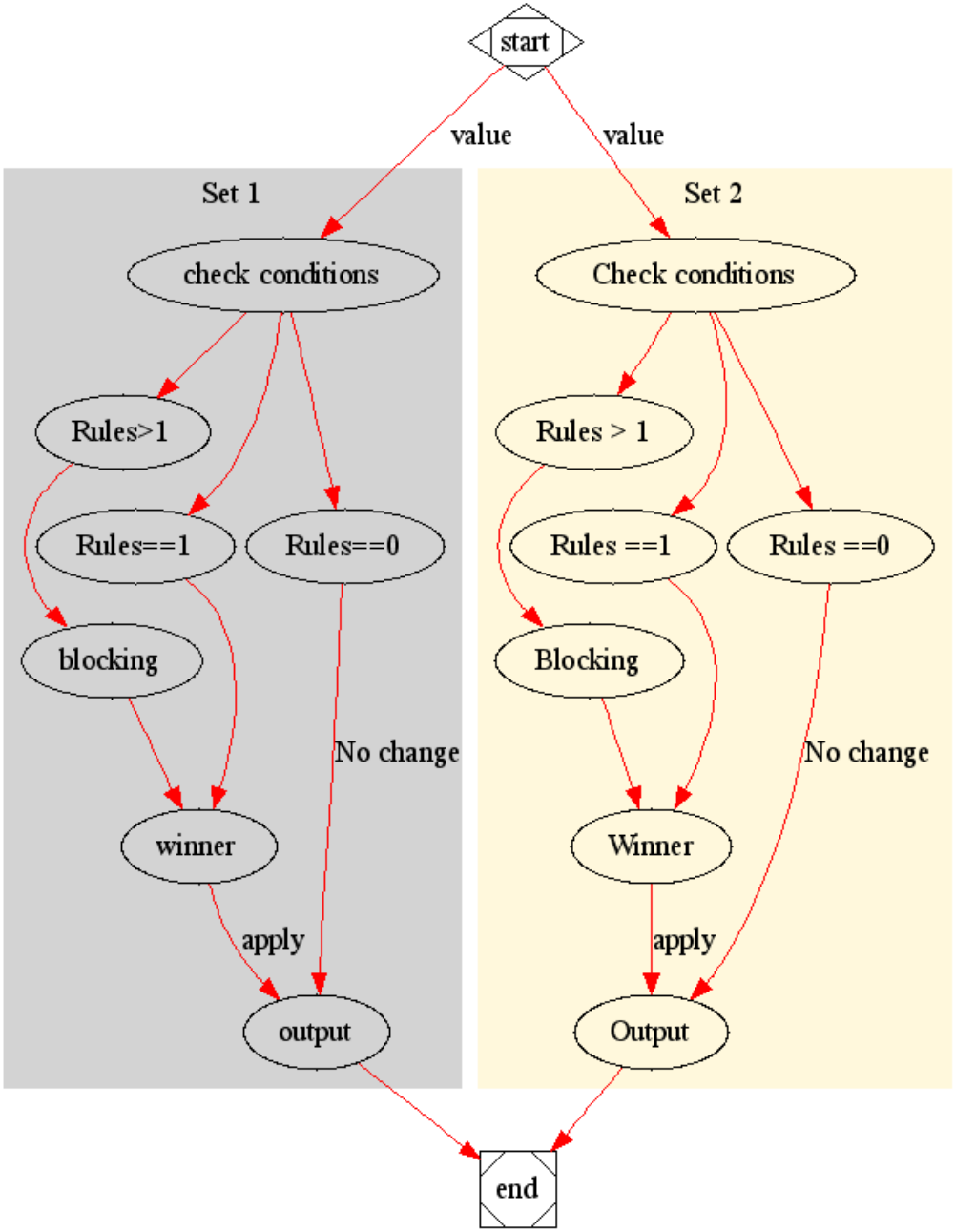


Figure : Blocking in *asiddhavat*

Pāṇini never said apply the rules in tripādī at the end sequentially.
He never said apply the rules in asiddhavat section parallely.
He used only three terms:
asiddhaH, asiddham, asiddhavat
asiddha: regarded as non-existing!
Variables local to a procedure are non-existent for the caller function.

Concept of different Data Spaces for different subroutines!
Global variables are accesible by all!

Data Encapsulation:

- Encapsulation of data with the (markers to the) functions
  - The *ñi* indicator indicates that such a root takes the suffix *kta* in the sense of present tense, as in

    $$ñidhṛṣā + kta \rightarrow dhṛṣṭa$$

  - Bhaj + (gh)a(ñ): In the presence of gh, j $\rightarrow$ g
- Multiple inheritance is $\rightarrow$ arranged as a linear inheritance derivational suffixes, deriving a noun from a noun (Taddhita pratyaya)
  (Ashwini Deo 2007)