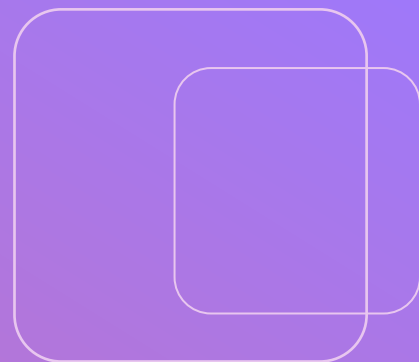




HEMIFUTURE PUBLIC SPECIFICATION

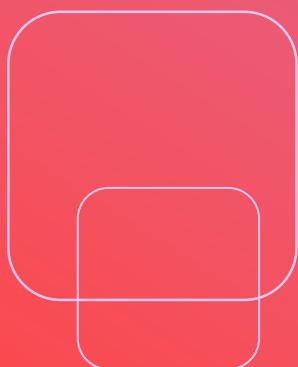
# Sevara 完整资产

语言伴读 / Typst 版

Current Greenfield Edition

Hemifuture

2026-07-10



让语言既可被说出，也可被权威执行。

*A language that can be spoken naturally and executed authoritatively.*

# 目录

|                                            |    |
|--------------------------------------------|----|
| 1. Sevara 是什么，它能做什么 .....                  | 5  |
| 1.1. 这本书怎么组织 .....                         | 5  |
| 2. 最小语言骨架 .....                            | 6  |
| 2.1. 三种句尾 .....                            | 6  |
| 2.2. 角色粒子 .....                            | 6  |
| 2.3. 代词与话语锚 .....                          | 6  |
| 2.4. 句段、范围与收束 .....                        | 7  |
| 2.5. 实现边界 .....                            | 7  |
| 3. 词库、构词与音形治理 .....                        | 7  |
| 3.1. 核心口语档 (Core Spoken Profile 1.0) ..... | 7  |
| 3.2. 词项身份 .....                            | 7  |
| 3.3. 构词闭包 .....                            | 8  |
| 3.4. 已登记派生 .....                           | 8  |
| 3.5. 复合词 .....                             | 8  |
| 3.6. 词形参考语言 .....                          | 8  |
| 3.7. 近音与口头审计 .....                         | 8  |
| 3.8. 当前最小核心词库 .....                        | 9  |
| 4. 标准例句与教学梯度 .....                         | 9  |
| 4.1. 三种话语模式 .....                          | 9  |
| 4.2. 目标与反身 .....                           | 9  |
| 4.3. 命名与重锚 .....                           | 9  |
| 4.4. 话题与关系从句 .....                         | 10 |
| 4.5. 条件与局部群体 .....                         | 10 |
| 4.6. 否定 .....                              | 10 |
| 4.7. 持续与终止 tail .....                      | 10 |
| 4.8. 长篇结界吟唱 .....                          | 10 |
| 4.9. 当前失败样章 .....                          | 11 |
| 5. 长篇组织与语义接口 .....                         | 11 |
| 5.1. 起始 stance frame .....                 | 11 |
| 5.2. 四类高负载边界 .....                         | 11 |
| 5.3. 低负载重锚 .....                           | 11 |
| 5.4. 条件分支 .....                            | 12 |
| 5.5. 关系从句 .....                            | 12 |
| 5.6. tail .....                            | 12 |
| 5.7. Surface 到 Canonical .....             | 12 |
| 5.8. Canonical 到 Execution .....           | 12 |
| 5.9. 推荐长篇模板 .....                          | 12 |
| 6. 语义、诊断与当前状态 .....                        | 13 |
| 6.1. 三层职责 .....                            | 13 |
| 6.2. 最早失败原则 .....                          | 13 |
| 6.3. 当前资产状态 .....                          | 14 |
| 6.4. 当前未闭合风险 .....                         | 14 |
| 7. 扩词、审查与维护手册 .....                        | 14 |
| 7.1. 发布前 clean break .....                 | 14 |

|                           |    |
|---------------------------|----|
| 7.2. 新词提交流程 .....         | 14 |
| 7.3. 派生词 card .....       | 15 |
| 7.4. 复合词 card .....       | 15 |
| 7.5. 名字、标题与 formula ..... | 15 |
| 7.6. 文档落点 .....           | 16 |
| 7.7. 当前最适合继续的任务 .....     | 16 |
| 7.8. 术语速查 .....           | 16 |

# 1. Sevara 是什么，它能做什么

## 阅读路径

本书从整体图景、最小示例和第一次可成立的话语开始展开。读者会先认识 Sevara 是什么、它能做什么、以及怎样进入实际使用；在此基础上，规则、术语和参考材料会按需要逐步展开。

Sevara 是一门受控自然语言式魔法语。它把“像语言一样可以被说出来”“像系统一样可以被稳定理解”“像规则一样可以被安全执行”三件事组织到同一门语言里。对读者来说，这意味着它既有自然表达的连续性，也有规范系统所要求的明确边界。

从功能上说，Sevara 至少能做三类事情。它可以陈述事实，例如“门在这里”；可以提出问题，例如“你打开门了吗”；也可以把一整段话送进施法执行流，例如“我为自己施加护障”或“若敌人进入火环，就束缚并焚烧他们”。你可以先把它理解成：同一套语言骨架，根据句尾不同，承担了三种不同用途。

门/通道 处于 在 此处域  
beror zore li hiran .

我 开启 目标 门/通道 提问  
mi harun vo beror ke

我 护持 目标 自己 施法  
mi talun vo ze ha

中文含义：

beror zore li hiran.: 门 / 通道位于此处域。

mi harun vo beror ke: 我是否开启目标门 / 通道？

mi talun vo ze ha: 我护持自身。

如果要把 Sevara 压缩成一句话来记，那就是：它是一门既能被玩家读写，又能被解析器归一化，并最终被权威执行端安全执行的语言。图 1 给出了这件事的整体轮廓。



图 1. Sevara 的三层结构。

这三层分别承担不同职责。自然表层（Surface Profile）负责让人真正能说出或写出句子；规范表层（Canonical Profile）负责让系统获得唯一、稳定的规范结果；执行配置（Execution Profile）则负责把通过检查的话语真正送入世界规则和资源预算之中。后面所有具体规则，不管是词尾、代词还是条件句，最后都要回到这三层关系上理解。

本书只教授 docs/sevara/spec/current.md 定义的当前语言。lexicon\_version = 6 只是仓库里可能暂时落后的工程数据编号；fixture、golden 与 demo 若和当前规范冲突，应该更新，而不能要求语言保留旧词形或旧解析路径。

## 1.1. 这本书怎么组织

这份完整资产采用“整体图景 → 最小语言骨架 → 词库与构词 → 例句 → 长篇结构 → 语义边界 → 治理手册”的顺序。第二章讲三种句尾、最短句骨架与基本语序；第三章讲音系、词项身份与构词；第四章用标准例句建立语感；第五章处理条件、关系从句、重锚与 tail；第六章集中说明 Canonical、Execution 和诊断边界；第七章给出扩词与审查清单。

如果你只想先学会读写，读到第五章已经足够；如果还要实现、审查或扩词，再继续读第六章和第七章。工程调试信息只属于 implementation\_reference.md，不再混入语言伴读正文。

## 2. 最小语言骨架

Sevara 的第一条规则不是“某个程序能不能跑”，而是“说话者是否给出了一段完整、可唯一解析的话”。当前语言只有一套 Surface；参考实现只能验证它，不能另立兼容语法。

### 2.1. 三种句尾

表 1. 同一门语言的三种话语模式。

| 句尾 | 言语行为 | 是否进入 Execution |
|----|------|----------------|
| .  | 陈述   | 否              |
| ke | 提问   | 否              |
| ha | 施法封印 | 是，由权威执行端复核     |

[stance frame,] [subject] predicate { role phrase } [tail] terminator

最短陈述可以只说明对象处于何处；最短施法通常需要主语、谓词、目标与封印：

门/通道 处于 位置 此处域 陈述  
beror zore li hiran .

我 开启 目标 门/通道 施法  
mi harun vo beror ha

### 2.2. 角色粒子

角色粒子放在补语之前，让同一个谓词可以明确区分目标、来源、位置、材料与名字。它们是闭合同，不从内容词派生。

表 2. 当前角色粒子。

| 粒子 | 角色                               | 示意                        |
|----|----------------------------------|---------------------------|
| vo | patient / target                 | harun vo beror            |
| ra | source / against / relative-to   | moren vo duron ra maren   |
| li | location / domain                | zore li hiran             |
| su | material / composition / bearing | beror su toran            |
| ne | naming complement                | renil vo beror ne "selan" |

### 2.3. 代词与话语锚

表 3. 当前代词。

| 词  | 职责                  | 边界               |
|----|---------------------|------------------|
| mi | 说话者 / 表层主语          | 不决定权威 caster 身份。 |
| ze | 当前 clause 显式主语的反身指称 | 没有说出的显式主语就不能绑定。  |
| ri | 最近安全话语锚             | 不得穿越高负载边界静默漂移。   |
| va | 条件或群体分支内的局部群体       | 分支退出后失效。         |

因此：

中文含义：

mi nerai vo ze ha 合法：ze 有显式主语 mi。

nerai vo mi ha 合法：匿名施法把显式 mi 用作目标。

nerai vo ze ha 非法：省略主语不会替 ze 创造词法锚。

## 2.4. 句段、范围与收束

ye 连接相邻 clause; sa 建立话题; ka 把关系从句限制在名词短语内部; no ... to ... 建立条件与结果; du / nu 只在主链结束后进入 tail。

stance -> topic / anchor -> main clauses -> condition / branch -> reanchor -> du / nu -> ha

自然感来自这些句段的节拍组织，而不是让听者猜省略了什么角色。只要命名、长关系从句、条件分支或群体分支提高了回指负载，就应先用一拍低负载句重锚，再继续新的高负载结构。

## 2.5. 实现边界

sevara\_demo 当前可能仍只认识旧工程数据。它的成功输出不证明一句话属于当前语言，它的失败也不能推翻本规范。语言资产先闭合；实现随后一次性切换到同一 contract，不保留 alias、fallback、legacy mode 或旧 fixture 验收。

# 3. 词库、构词与音形治理

Sevara 采用 lexicon-first。解析器先识别精确登记的词项，再进入句法；拼写看起来合规、词尾相似或语义似乎合理，都不能创建词项身份。

## 3.1. 核心口语档（Core Spoken Profile 1.0）

表 4. 核心字母的规范教学音值。

| 字母组         | 音值            | 字母组               | 音值                     |
|-------------|---------------|-------------------|------------------------|
| a e i o u   | /a e i o u/   | b d g h k l m n p | /b d g h k l m n p/    |
| r s t v y z | /r s t v j z/ | w / f             | /w/、/f/，只在精确接受词项中局部许可。 |

普通音节按 (C)V(V)(C) 切分。单个元音间辅音进入后一音节的 onset；两个元音间辅音按 coda.onset 分开。普通多音节词默认首音节主重音；闭合粒子通常弱读，但在 reanchor、frame 或 tail 边界可取次重音。

表 5. 当前相邻元音的读法。

| 序列 | 读法                         | 精确词项                  |
|----|----------------------------|-----------------------|
| ai | /a <sub>i</sub> /，同音节下降复元音 | nerai / mirai / sorai |
| ei | /e <sub>i</sub> /，同音节下降复元音 | mei                   |
| ae | /a.e/，hiatus               | wae                   |
| ou | /o.u/，hiatus               | fou                   |
| ao | /a.o/，hiatus               | vao                   |

这些读法是精确行事实，不能类推出一套新词模板。token 之间不吞音、不跨词重切、不插桥接音；停顿、旋律、拖音与手势都不能替代实际 token。

## 3.2. 词项身份

当前只承认两种状态：

表 6. 词项状态。

| 状态                | 语言地位   | 解析行为                                 |
|-------------------|--------|--------------------------------------|
| accepted_registry | 当前合法词项 | 可创建明确的 Surface 与 Canonical identity。 |
| proposal          | 候选材料   | 不得进入解析、教学正例或可执行 Canonical。           |

工程数据版本、历史词表、草案、例句和 parser 行为都不能自动把 proposal 提升为 accepted\_registry。

### 3.3. 构词闭包

当前构词只有四条入口：

表 7. 构词入口。

| 入口                | 最低证据                                                      | 结果                  |
|-------------------|-----------------------------------------------------------|---------------------|
| 精确基础词项            | lexeme_id + exact surface + class + profile               | 登记后成为独立词项。          |
| 已登记派生             | rule_id + accepted base + output surface + semantic delta | 只接受列出的输出，不开放类推。     |
| 已登记复合             | 组件身份、关系、边界、输出表面与近音审计                                      | 只接受完整 compound row。 |
| 名字 / 标题 / formula | 专属 card、使用域与 authority guard                              | 只在声明的 span 与语境中成立。  |

exact lexeme -> registered derivation -> registered compound -> reviewed proposal -> reject

硬边界：

- 任何末尾元音、profile suffix 或 cadence family 都不能自动决定词类或生产合法词。
- root 只保存词源、音形亲缘与造词动机，不是可直接说出的词。
- beror、harun、duri 的末尾字母都是各自词项的一部分，不是开放后缀。
- 未登记复合词不因组件都存在而合法；未登记派生词不因规则“看起来适用”而合法。

### 3.4. 已登记派生

派生规则必须声明输入类别、输出类别、语义变化、音形变化与可枚举输出。规则本身可以复用，但每个输出仍需进入 registry；这使构词保持可教、可审计，也防止 suffix 重新变成暗中的开放生产类。

实验性 derived stem family 可继续作为 proposal 生成器，但 stem mold、cadence family 和 class signal 只帮助审查节拍与语义厚度。只有逐项升格的 surface 才能成为正式词项。

### 3.5. 复合词

复合词优先表达可分析的新概念，但必须满足：

1. 每个组件已有 accepted identity。
2. 组件关系明确，不依赖听者猜测省略关系。
3. 组件边界可从连续语流中恢复，不靠异常停顿或重音。
4. 输出表面完成近音、同形、教学与长篇行读回审计。
5. 完整输出有独立 registry row。

组件删除、临时桥接音、不可恢复的 sandhi 与“为了华丽而换尾”都应拒绝；必要时退回短语表达。

### 3.6. 词形参考语言

外部语言只提供音形参考，不提供语法、专名、脚本或 registry 捷径。每个新造候选必须记录：

reference language + source word + source meaning + derivation steps + borrow risk

北欧 / 日耳曼语言可提供短功能词的辅音骨架；Quenya / Sindarin 可提供清晰音系、液音和高语域节奏；Hymmnos 只可提供咏唱分段与表演层启发。不得复制专名、标志性短语、歌词、脚本或自然语言屈折范式。

docs/sevara/lexicon/high\_load\_surface\_replacement\_candidates.md 当前只是候选账本。候选在 replacement gate 接受前不是 Sevara 词；一旦接受，被替换形式同轮从 active spec、active lexicon、教学资产和实现输入中删除，不保留别名。

### 3.7. 近音与口头审计

每个候选至少提交 phonemic\_form、syllabification、primary\_stress、nearfield\_compare\_cohort、孤立读回与代表性行读回。高负载闭合同词必须在同类、跨类和长篇邻接位置中比较。

表 8. 最小口头 gate。

| 检查 | 阻塞条件                   | 处理                     |
|----|------------------------|------------------------|
| 同形 | 与当前词项或高频名字同形           | 拒绝或重造。                 |
| 近音 | 高负载集合中只有一个可辨音位差        | 默认阻塞。                  |
| 语流 | 连读时边界消失或稳定滑向另一词        | 重造，不用异常停顿补救。           |
| 教学 | 同一课或同一长篇中持续混淆          | 拆批仍不够时重造。              |
| 审美 | 拼写与读法虽可解析但破坏整体 palette | 回到 source ledger 重新派生。 |

3.8. 当前最小核心词库

表 9. 当前词库的教学分组。

| 分组       | accepted surface                                                     |
|----------|----------------------------------------------------------------------|
| 代词       | mi / ze / ri / va                                                    |
| 范围与结构    | ye / no / to / sa / ka / na / du / nu / ha                           |
| 角色       | vo / ra / li / su / ne                                               |
| stance   | wae / fou / irel / ayel / mei / vao / rul                            |
| 基础操作     | harun / talun / nerai / renil / selka / nore / mirai / moren / zore  |
| 对象、状态与时间 | beror / duron / duri / maren / hiran / toran / soran / sorai / vesan |
| 质料与复合力   | livu / gemor / nimlumu / bomil / talum / bomtalum / livkorum / korar |

quoted teaching name "selan" 是稳定教学锚，不证明真名、所有权、对象存在或权限。

4. 标准例句与教学梯度

本章只展示当前语言。正例可以直接作为教学材料；反例只展示当前规则为何拒绝某个结构，不承担历史版本对照。

4.1. 三种话语模式

中文含义：

beror zore li hiran.: 门 / 通道位于此处域。

mi harun vo beror ke: 我是否开启这道门 / 通道？

mi harun vo beror ha: 令我开启这道门 / 通道。

. 与 ke 不进入 Execution；ha 才把整段话送入权威执行流。

4.2. 目标与反身

表 10. 显式目标与反身锚。

| 表面                | 判定 | 原因                       |
|-------------------|----|--------------------------|
| mi talun vo ze ha | 合法 | ze 回到当前 clause 的显式主语 mi。 |
| mi nerai vo ze ha | 合法 | 治疗目标是显式主语自身。             |
| nerai vo mi ha    | 合法 | 匿名施法，但目标 mi 已明确说出。       |
| nerai vo ze ha    | 非法 | ze 没有说出的显式主语锚。           |

4.3. 命名与重锚

mi renil vo beror ne "selan" ha

ne 只引入命名补语；"selan" 是教学名，不是真名证明。命名后若要进入条件、群体、长关系从句或高负载动作链，先用低负载句把名字立回当前拍：

mi renil vo beror ne "selan", ye "selan" zore duri

## 4.4. 话题与关系从句

sa beror ka zore duri, mi harun vo ri

sa 建立话题，ka zore duri 只限制该话题 head。后续 ri 回到最近安全锚点；关系从句内部出现的其他对象不会自动抢走外层焦点。

fou ayel mei, mi mirai vo beror ka zore bomil li hiran ha

这句话从起始 stance frame 进入主句，ka zore bomil li hiran 只描述目标门 / 通道。它不证明目标存在、可见、可达或允许显现。

## 4.5. 条件与局部群体

no maren nore li "selan", to selka vo va

no 建立条件，to 建立结果。va 只承接当前分支中的局部群体；分支退出后若还要回到 "selan"，应先显式重锚。

## 4.6. 否定

表 11. 当前否定位置。

| 表面                                          | 作用                |
|---------------------------------------------|-------------------|
| mi na harun vo beror ha                     | na 否定 harun。      |
| no maren na nore li "selan", to selka vo va | na 否定条件前件中的 nore。 |

na 固定在所否定谓词之前，不移动到 clause 尾部，也不作为词项后缀。

## 4.7. 持续与终止 tail

mi talun vo "selan", du vesan, nu sorai ha

du vesan 提出持续时段，nu sorai 提出终止界。tail onset 后不得再追加主链；资源是否足够、黎明是否到达和效果是否保持都由权威执行端判断。

## 4.8. 长篇结界吟唱

fou ayel vao,  
sa beror su toran ka zore bomil li hiran,  
mi renil vo ri ne "selan", ye "selan" zore duri,  
no maren nore li "selan", to selka vo va,  
ye mirai vo "selan", ye talun vo "selan",  
du vesan, nu sorai ha

这条长篇按六拍组织：

1. 起始 stance 单独成拍。
2. sa ... ka ... 建立长话题。
3. renil 命名后立刻低负载重锚。
4. 条件分支只在自己的 frame 内使用 va。
5. 主链动作使用明确的 quoted name，避免让 ri 穿越分支漂移。

6. du / nu / ha 只在所有主链完成后收束。

## 4.9. 当前失败样章

表 12. 失败与修复。

| 失败表面                                                  | 失败点                     | 修复                                                    | 理由                       |
|-------------------------------------------------------|-------------------------|-------------------------------------------------------|--------------------------|
| nerai vo ze ha                                        | E_BINDER_MISSING        | mi nerai vo ze ha                                     | 显式说出 ze 的主语锚。            |
| mi renil beror "selan" ha                             | role frame 缺失           | mi renil vo beror ne "selan" ha                       | 目标与命名补语都必须显式标记。          |
| mi talun vo "selan", du vesan, ye mirai vo "selan" ha | tail onset 后回扩主链        | mi talun vo "selan", ye mirai vo "selan", du vesan ha | 先完成主链，再进入 tail。          |
| fou, mi mirai vo berorE ha                            | INCOMPLETE_STANCE_FRAME | fou ayel mei, mi mirai vo beror ha                    | stance frame 缺少 emotion。 |

合法形式还不等于施法成功。名字是否指向对象、目标是否可达、材料是否存在、权限与预算是否通过，始终属于 Execution。

## 5. 长篇组织与语义接口

Sevara 的长篇不是把短句无限相接，而是把信息负载分配到可听辨的拍点。每一拍都应让听者知道当前对象、当前范围和接下来要进入的结构。

### 5.1. 起始 stance frame

stance emotion [intensity],

当前 stance words 为 wae / fou, emotion words 为 irel / ayel, intensity words 为 mei / vao / rul。stance frame 只出现在整段话开头，不在 clause chain 中间重开，也不放到 ha 前临时加码。

stance 可以影响 Execution 的风险、倍率、冷却或预算参数，但不能创建动作、目标、材料、持续时间或权限。

### 5.2. 四类高负载边界

表 13. 长篇中需要分拍的边界。

| 边界                 | 风险                    | 推荐处理                    |
|--------------------|-----------------------|-------------------------|
| 长 sa ... ka ... 退出 | 关系从句内部材料争夺外层焦点        | 先用 ri 或显式 NP 重启外层主链。    |
| renil 命名完成         | quoted name 尚未成为稳定话语锚 | 插入 "name" zore duri 一拍。 |
| 条件 / 群体分支退出        | va 的局部群体与外层对象竞争       | 先显式恢复外层对象，再开新链。         |
| tail onset         | 听者误以为主链已经结束           | 主链全部前移；du / nu 之后只允许收束。 |

### 5.3. 低负载重锚

重锚不是冗词，而是口头语言里的结构标点。当前推荐的低负载形包括：

中文含义：

"selan" zore duri: 名字对象处于稳定 / 封闭态。

beror zore li hiran: 门 / 通道重新落回此处域。

mi harun vo ri: 以最近安全锚为明确目标重启主链。

若一句话只有在作者心里才能知道 *ri* 指向谁，就不应要求解析器或听者猜测。改用 *quoted name*、显式 NP 或一拍状态句。

## 5.4. 条件分支

no Condition, to Result

条件 *frame* 是局部作用域。前件只描述条件；后件只在该 *frame* 内承接 *va* 等局部锚。条件是否真实、对象是否属于群体、权限是否允许结果动作，都不是语言层结论。

连续条件不应争用同一拍。前一分支结束后若要回到外层对象，应先重锚，再开始下一条 *no ... to ...*。

## 5.5. 关系从句

*ka* 只修饰紧邻的 NP *head*。嵌套深度最多为 2；关系从句退出后不得把内部对象静默提升为外层 *ri*。长 *ka* 后接条件或群体分支时，先用一条低负载 *outer clause* 重启 *head*。

## 5.6. tail

*du* 表示 *duration tail*，*nu* 表示 *terminus tail*。它们只在 *ha discourse* 中投影执行语义，且最多各出现一次：

main clause chain, *du* Duration, *nu* Terminus *ha*

同时出现时，语言只表达两个界；具体以哪个更早结束、能否维持和资源如何消耗，由 *Execution* 决定。

## 5.7. Surface 到 Canonical

归一化按固定顺序进行：

exact registry lookup -> mode -> stance -> clause/scope -> subject -> roles -> ze/ri/va -> type -> tail -> authority questions

前一步失败时，后一步不得用语义常识或世界状态修补。*Canonical* 只显式化已经说出的结构，并记录 *subject\_provenance* = *explicit* | *restored\_default*；它不加入隐藏目标、隐藏材料或隐含 *permission*。

## 5.8. Canonical 到 Execution

只有 *ha* 话语进入权威执行环境。*Execution* 至少重新检查：

- *caster* / *source* 的权威身份；
- *target*、*name* 与 *world object* 的解析；
- *permission*、*visibility* 与 *reachability*；
- *resource*、*budget* 与 *lifecycle*；
- *effect risk*、冲突和原子提交条件。

客户端解析结果、UI 选择和本地缓存都不能成为权威。*Execution* 的成功或失败也不能反向改写词项身份、句法角色或 *Canonical* 文本。

## 5.9. 推荐长篇模板

stance,  
topic + relative clause,  
main action + naming,  
low-load reanchor,  
condition + local branch,  
explicit outer reanchor,  
remaining main actions,  
duration / terminus + *ha*

这个模板不是强制的诗节数量，而是一条负载梯度：先立世界与对象，再做动作；先闭合局部分支，再恢复外层连续性；最后才收束时间与执行模式。

## 6. 语义、诊断与当前状态

### 6.1. 三层职责

表 14. Surface、Canonical 与 Execution 的职责。

| 层         | 输入                | 输出                                               | 禁止事项                      |
|-----------|-------------------|--------------------------------------------------|---------------------------|
| Surface   | 说出或写下的当前 Sevara   | 唯一可分词、可分类的句段                                     | 用后缀、近音或历史词形猜词。            |
| Canonical | 通过语言检查的 Surface   | 确定的 clause、role、scope、tail 与 authority questions | 写入 world truth、执行成功或隐藏目标。 |
| Execution | Canonical + 权威上下文 | SpellPlan 或拒绝结果                                  | 相信客户端身份、缓存或本地世界状态。        |

### 6.2. 最早失败原则

诊断应报告语言层最早失败点。词项未登记时，不继续猜词类；role 缺失时，不继续猜目标；ze 缺锚时，不继续生成 Canonical；语言通过后，world guard 的拒绝也不能伪装成语法错误。

表 15. 当前公开诊断的核心子集。

| 代码                        | 级别      | 含义                                 |
|---------------------------|---------|------------------------------------|
| E_LEXEME_UNREGISTERED     | Error   | 词形未命中 exact accepted registry row。 |
| E_BAD_WORD_SHAPE          | Error   | token 不符合当前字符与音形约束。                |
| E_BINDER_MISSING          | Error   | ze 或分支代词缺少合法锚。                     |
| E_RI_UNRESOLVED           | Error   | ri 没有最近安全锚。                        |
| E_RI_AMBIGUOUS            | Error   | ri 有多个同优先级候选。                      |
| E_NAME_COMP_REQUIRED      | Error   | 命名谓词缺少 ne quoted_name。             |
| E_NAME_REBIND             | Error   | 同一 discourse 中名字被绑定到不同对象。          |
| E_TYPE_MISMATCH           | Error   | role 已形成，但谓词配价与补语类型不匹配。            |
| E_STANCE_POSITION         | Error   | stance frame 不在 discourse 起始。      |
| E_INCOMPLETE_STANCE_FRAME | Error   | stance 或 emotion 成分缺失。             |
| E_DU_POSITION             | Error   | du 不在 cast tail。                   |
| E_NU_POSITION             | Error   | nu 不在 discourse 末尾的 cast tail。     |
| W_STANCE_TYPE_MISMATCH    | Warning | stance 与谓词域低匹配，但不改变语言合法性。          |
| W_HIGH_STANCE_RISK        | Warning | Execution 可能提高风险。                  |

结构化诊断至少稳定提供 level + code。message、position、expected、actual 与 suggestion 可以扩展，但不得改变错误归类。

### 6.3. 当前资产状态

表 16. 当前事实来源与状态。

| 资产                                        | 地位       | 要求                                       |
|-------------------------------------------|----------|------------------------------------------|
| docs/sevara/spec/current.md               | 语言规范     | 决定合法 Surface、Canonical 形状与 Execution 边界。 |
| docs/sevara/complete_asset/current.typ    | 教学伴读     | 只能解释当前规范，不得另立规则。                         |
| docs/sevara/lexicon/ 当前 accepted registry | 词项身份     | 必须与规范和教材同轮更新。                            |
| bootstrap_minimal.json / Rust demo        | 待对齐的工程实现 | 若仍承载已删除规则，应直接重构，不保留兼容入口。                 |
| fixture / golden                          | 验证资产     | 只验证当前 contract；旧样例不再是 acceptance test。   |
| history / review / nightly                | 研究证据     | 不参与当前解析与验收。                              |

### 6.4. 当前未闭合风险

语言层已明确删除开放 suffix production，但工程 registry、fixture、golden 与 Rust resolver 仍需逐项审计并一次性切换。完成切换前，demo 只能当作旧工程现状观察工具，不能当作当前语言教学器；词库还需要为已接受词项补齐统一的 lexeme\_id、音系字段、词类、配价、近音比较集合与 source ledger；构词层还需要把 registered derivation 与 registered compound 的 card schema 收成可审查的最小闭集。这些风险不产生兼容义务。正确的收口方式是修资产与实现，使它们服从当前规范。

## 7. 扩词、审查与维护手册

### 7.1. 发布前 clean break

Sevara 尚未公开发布。新设计一旦被接受，就直接成为唯一 current contract：

accept replacement -> update active registry/spec/teaching/implementation -> delete replaced form -> regenerate current fixtures

不得保留 alias、fallback、legacy parser mode、旧 fixture 验收或双词表。历史材料可以留在 history / review 中解释为什么做出决定，但不能进入 current lookup。

### 7.2. 新词提交流程

表 17. 新词从来源到接受的流水线。

| 阶段              | 必须回答                                                  | 不通过时           |
|-----------------|-------------------------------------------------------|----------------|
| 语义需求            | 现有词、短语、派生或复合为何不足？                                     | 继续使用现有表达。      |
| source ledger   | 参考语言、source word、source meaning、derivation steps 是什么？ | 退回补证据。         |
| 音系设计            | 音位、音节、重音、语流和口头节拍如何？                                   | 重造。            |
| nearfield audit | 与高负载词、同类词、名字和长篇邻接项是否混淆？                               | 阻塞或重造。         |
| 形态身份            | 基础词、登记派生、登记复合还是专属 card？                               | 不得进入 registry。 |
| 句法与语义           | 词类、配价、role、scope 与 authority guard 是什么？               | 退回设计。          |
| 教学验证            | 是否有孤立读回、最小句、长篇行读回与失败对照？                               | 保持 proposal。   |
| 接受              | active spec、registry、教材与实现是否同轮一致？                     | 不得宣布 current。  |

### 7.3. 派生词 card

已登记派生至少需要：

- derivation\_rule\_id
- base\_lexeme\_id
- output\_lexeme\_id
- exact\_surface
- input\_class / output\_class
- semantic\_delta
- phonological\_effect
- nearfield\_compare\_cohort
- teaching\_example
- authority\_guard\_required

规则可以描述一组候选，但 registry 必须逐项列出接受输出。没有 output row，就没有词项。

### 7.4. 复合词 card

已登记复合至少需要：

- compound\_id
- component\_refs
- compound\_relation
- exact\_surface
- component\_boundary
- semantic\_profile
- word\_class
- valency\_or\_attachment
- nearfield\_compare\_cohort
- teaching\_example
- authority\_guard\_required

复合词不能通过删音、桥接音、异常重音或自由换尾掩盖组件边界。若短语已经清楚，优先保留短语，不为了词表规模强造单词。

### 7.5. 名字、标题与 formula

quoted name、ritual title、true name 与 formula 必须使用各自 card，不得借普通 lexeme 的外形偷取身份。

教学名应避免：

- 高频闭合词同形；
- 当前 accepted lexeme 同形；
- tail anchor 同形；
- “现有词项 + 一个音位”的近似形；
- 同一课内只有一个可辨音位差的名字对。

"selan" 只是一条教学名，不证明对象存在、所有权、真名或控制权。

## 7.6. 文档落点

表 18. 不同信息的唯一落点。

| 问题              | 写入位置                                    |
|-----------------|-----------------------------------------|
| 语言规则是什么？        | docs/sevara/spec/current.md             |
| 词项与构词 card 是什么？ | docs/sevara/lexicon/                    |
| 怎样教学与长篇朗读？      | docs/sevara/complete_asset/ 与 examples  |
| 实现怎样承接、怎样调试？    | docs/sevara/implementation_reference.md |
| 这轮为什么改、接下来做什么？  | backlog.md、roadmap.md、dev report        |
| 旧设计为什么曾经存在？     | history/、reviews、nightly                |

## 7.7. 当前最适合继续的任务

1. 给当前 accepted lexicon 建立单一 registry 数据源，补全每项的音系、词类、配价、source ledger 与 nearfield cohort。
2. 冻结 registered derivation / compound card schema，并用少量真正需要的新词验证它，而不是再造一套 suffix family。
3. 一次性重写 parser lookup、fixture 与 golden，使其只接受当前 Surface，并删除已经失效的诊断码和旧解析分支。

## 7.8. 术语速查

表 19. 当前术语。

| 术语                    | 中文说明                                             | 边界                    |
|-----------------------|--------------------------------------------------|-----------------------|
| Surface               | 说出或写下的语言表层                                       | 只包含当前 accepted forms。 |
| Canonical             | 确定性归一化结果                                         | 不包含 world truth。      |
| Execution             | 权威世界中的执行计划与结果                                    | 客户端不权威。               |
| accepted_registry     | 当前合法词项或 card                                     | 精确匹配，不模糊补全。           |
| proposal              | 待审候选                                             | 不创建 identity。         |
| root                  | 词源与音形亲缘骨架                                        | 不是可直接说出的词。            |
| registered derivation | 逐项登记输出的派生                                        | 规则不开放生产。              |
| registered compound   | 逐项登记的复合词                                         | 组件存在不等于输出合法。          |
| role particle         | 标记 target、source、location、material、name 的<br>闭合词 | 不证明 world relation。   |
| reanchor              | 重新建立安全话语焦点的低负载拍                                  | 不是冗余补词。               |
| authority question    | 语言层交给执行端回答的问题                                    | 不是 answer 或 success。  |