```
                              (get input $)
                                   │
                                   ▼
                              Word list
                                   │
        Separate by [then .]       │
                                   │
    ┌──────────┬──────────┬────────┴────────┬──────────┐
    ▼          ▼          ▼                 ▼          ▼
Word list  Word list  Word list         Word list  •••  Word list
               │
               │    Consider one at a time
               ▼
         *(understand $Words as $ComplexAction)
               │
At least one solution?   │   Yes
               ▼                        No, separate by [and ,]
         Complex action
               │
               │   Sieving based on likelihood
               ▼
         Complex action
               │
               │   Interactive disambiguation
               ▼
         Complex action
               │
               │   Enumeration
    ┌──────────┼──────────┐
    ▼          ▼          ▼
  Action     Action  ••• Action
    │          │          │
    ▼          ▼          ▼
 (try $)    (try $)    (try $)
```

Program entry point

Intro

Read-parse-act loop

Prompt & get input

Top-level parser (parsing and acting are interleaved)

Action parser

Try

Tick

Action-handling predicates

Responses, cutscenes, and updates to the world

Tick handlers

Game-over loop

Object parser

Topic parser

Amusing

Status bar

World model:

Rooms and other **objects** with names and synonyms, traits, descriptions, map connections, initial locations, flags, properties...

Directions and relations

Standard traits

Rules defined by the standard library

Rules defined by the story author

Rules defined by the standard library and, optionally, the story author

Standard library

- Framework
- Parser
- Basic world model
- Standard actions
- Utilities

`.dg`

Story

- Narrative
- Objects and map
- Action handlers
- Metadata

`.dg`

Queried by

Embedded media and **cover art**

`.png/.jpg`

Compile

Compile

Include

Internal representation

Optimize

Generate

Generate

Story file (Z-code)

`.zblorb/.z8/.z5`

Story file (Å-code)

`.aastory`

Bundle

Bundle

Disk image (Commodore 64)

`.d64`

Web directory

`.html/.css/.js`

Execute

Execute

Execute

Z-machine Interpreter

Vintage hardware

Web browser

```
Foyer of the Opera House
You are standing in a spacious hall, splendidly decorated in red
and gold, with glittering chandeliers overhead. The entrance from
the street is to the north, and there are doorways south and west.

> i
You have no possessions. You're wearing a velvet cloak.

> (now) (#hook is #heldby #player)
> i
You have a small brass hook. You're wearing a velvet cloak.

> i
The source code has been modified. Merging changes into the
running program.
You have a small brass hook. You're wearing a flamboyant yellow
hat and a velvet cloak.

>
```
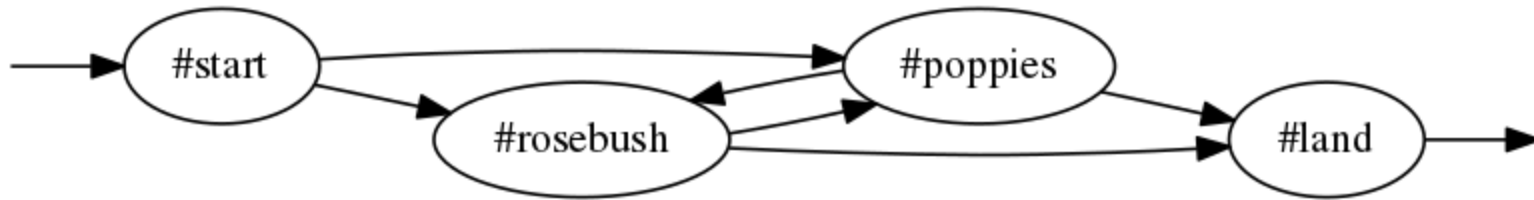
(try $)

(refuse $) | *(before $) | (refuse $) | (instead of $)

(prevent $) | (perform $) | *(after $)

Action-specific predicates
e.g. (descr $Obj) or
(narrate leaving $Room $Dir)

Time ⟶

[sit]　　　　[push $Obj]　　　　[find $Obj]

⋮ On what?　　　⋮ In what direction?

[climb $Obj]　[push $Obj $Dir]　[go to $Room]　[leave]　[stand]

[leave $Room $Dir with $Obj] ⤸　[go $Dir]　[leave $Obj]

[leave $Room $Dir] ⤸　[leave $Room $Dir by $Vehicle] ⤸
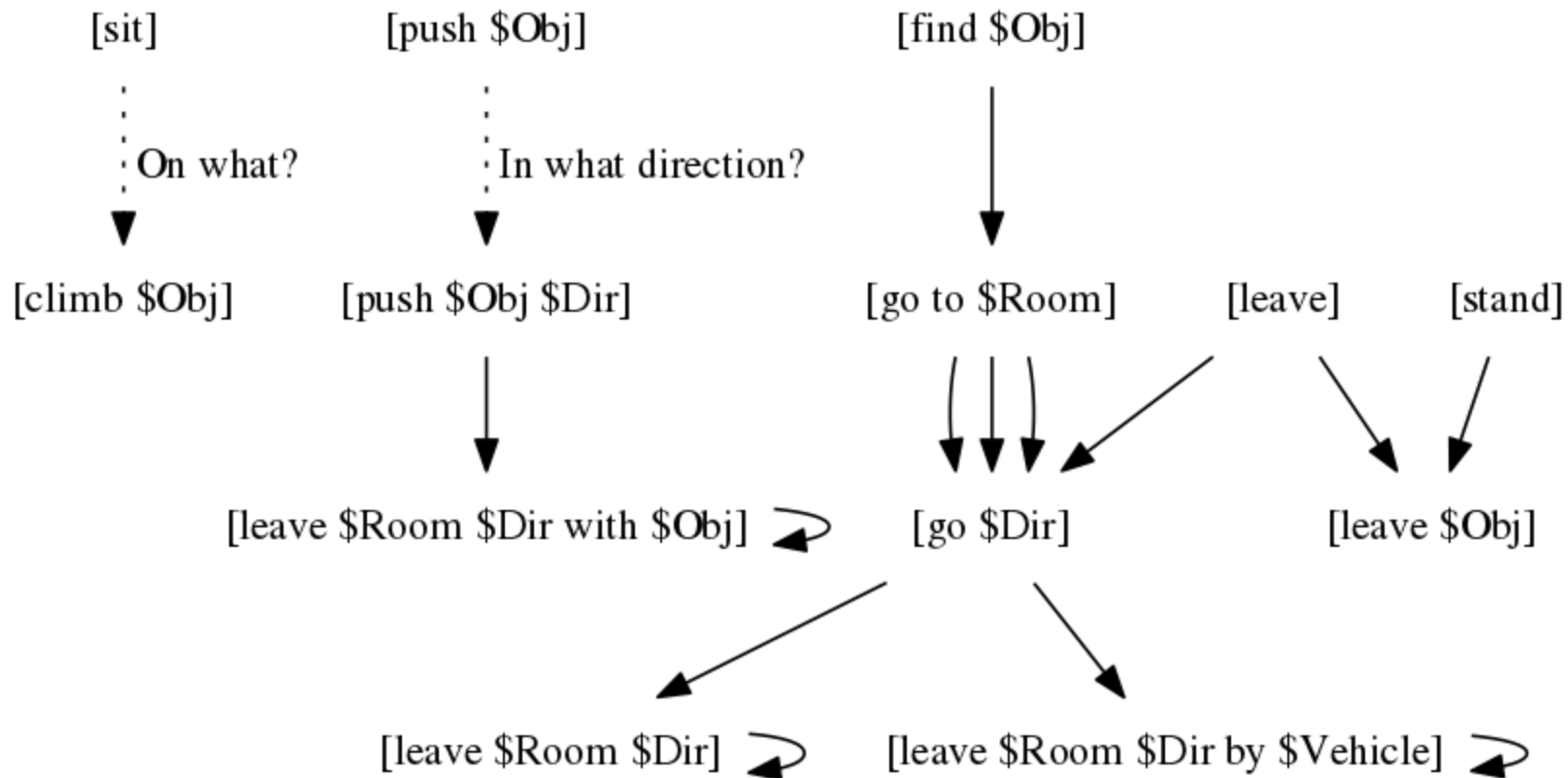
(excluded from all $)

(out of reach $)   (not here $)   (fine where it is $)   (intangible $)   (topic $)
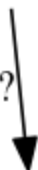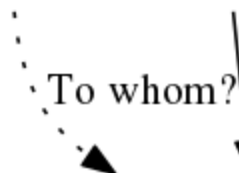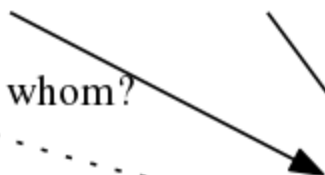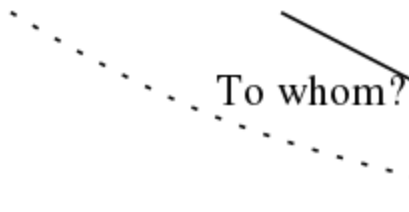
(container $)   (supporter $)

(actor container $)   (seat $)   (actor supporter $)

(room $)   (in-seat $)   (on-seat $)

(door $)   (vehicle $)   (direction $)   (relation $)

(opaque $)

(animate $)　　　(openable $)　　(item $)

(female $)　(male $)　　(lockable $)　　(wearable $)

(potable $)　(edible $)　(sharp $)　(consultable $)

(pushable $)　(switchable $)