Explanation

This is showing that for all the system states that can exist, there is an operation to exit that state and that all states can enter the exit condition, which allows wov-push to work.

The system states (right) show what is impacted by db changes which needs to be managed. The operations (right) show the tools that read system state (selection matrix) and alter the system (operation matrix).

System States

1. db schema changes[1]
2. delta files exist
3. git repo changes
4. registry delta changes for pver
5. registry git changes

Operations

A. db schema changes (ex. cli)
B. user adds deltas (wov-db-edit)
C. commit deltas to registry (wov-db-commit)
D. exit! (wov-push allowed by wov-pushcheck-db)
E. git commit on registry

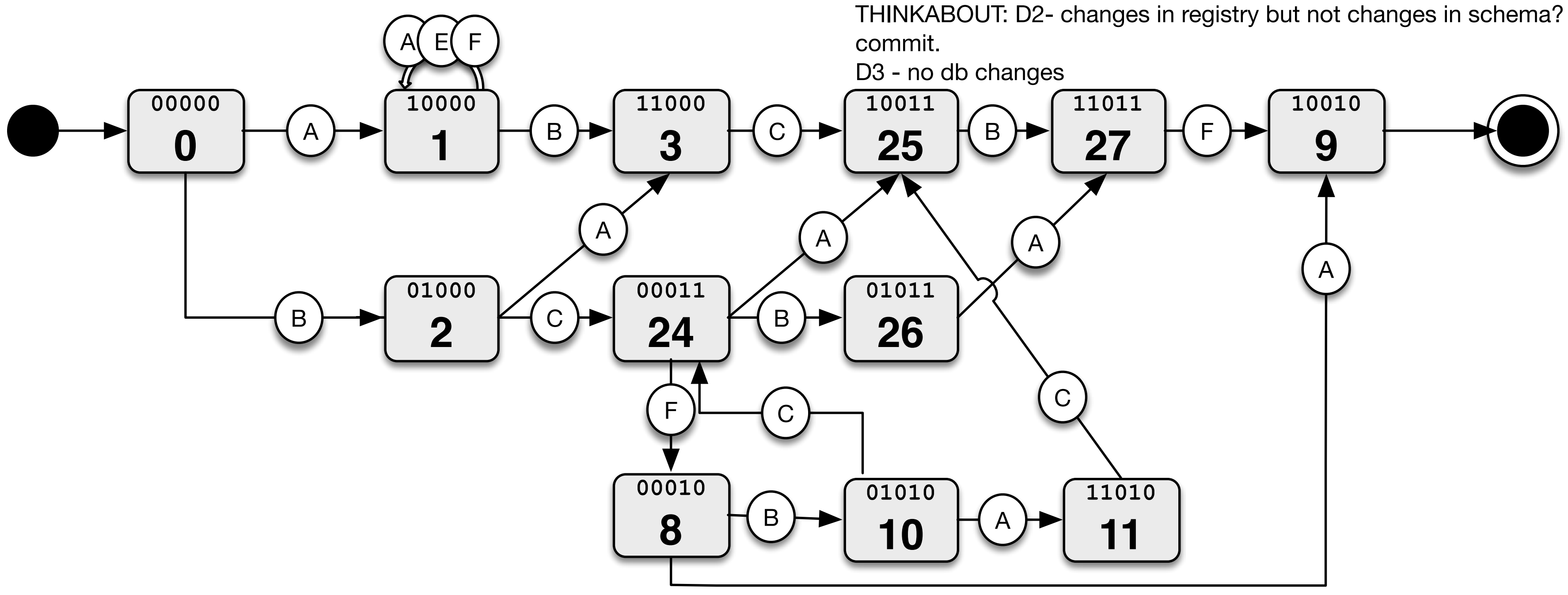
```
Selection Matrix
   A B C D   E
  ┌-------------
1 |*|  |  |TFF| |
2 |  |*|T|FFF| |
3 |  |  |  |FFF| |
4 |  |  |  |TTF| |
5 |  |  |  |FFF|T|
```

```
Operation Matrix
   A B C D E
  ┌----------
1 |T|  |  |E| |
2 |  |T|F|X| |
3 |  |  |  |I| |
4 |  |  |T|T| |
5 |  |  |T|!|F|
```

[1] check schema changes by comparing checksum of dbschema dump to registry's checksum for the parent db

NOTE: Hmm, if there is an entry in registry, for the current PVER, then we have already pushed and are adding to the db. Ok, since that is assumed… but there will be a 'case' where code was checked into archive and db changes were iterated on…. probably need a db number like secrets number…

D1- changes accounted for in registry
THINKABOUT: D2- changes in registry but not changes in schema? 1st commit.
D3 - no db changes



Files

wovtools/db/[dbname].deltas
- here, each database stores its changes from the parent

wovtools/db/registry
- this is a git archive containing the schemas of one or more databases.

wovtools/db/registry/[dbname].json
- The schema for the database, used to track parents, snapshots and checksums.

wovtools/db/registry/[dbname]/[pver].deltas
- All the changes to a database for a pver.

Example for 'mydatabase'
wovtools/db/mydatabase.deltas
wovtools/db/registry/mydatabase/[pver].deltas
wovtools/db/registry/mydatabase.json

Schema Format (json)

```
{
  "//wovtoolsversion" : "for matching file formats",
  wovtoolsversion : X,

  "//versions" : "stores db version info",
  versions: {
    [pver] : {
      checksum: "$(wov-db-checksum [dbname] | shasum -a 256)"
      "//parent" : "stores a reference to parent, with space for
managing parents differently",
      parent : {
        pver: [version],
        ~~type : [pver|snapshot],~~
        ~~snapshot : { TBD }~~
      }
    }
  }
}
```

Commands

wov-db-edit [dbname]
- opens an editor to enter the database specific schema changes
- ex. wov-db-edit mydatabase edits wovtools/db/mydatabase.deltas

wov-db-commit
- purpose is to move changes for a pver into the registry for versioning.
1. if all main repo git checked in, continue (this 'seals the database with the code').
2. move deltas into registry
3. update db registry entry for this pver
4. update registry database pver number, with checksum

wov-pushcheck-db
- purpose is to make sure database is all checked in, before allowing wov-push to occur
- Two states of system can pass
  - if all false
  - or, if db schema changes (1) then there are registry delta changes for pver (4) to account for the db schema changes, while rest false, meaning it is all checked in
    ?? how about db with no parent (first commit) as it could potentially not differ from checksum ??