* - no reserved words in factotum, just some symbols
* (p 12)
* **Marker:** 
  + Beginning a fact, symbols: (first char)
  + Indicates to Factotum how to treat this fact [line in the fact??]
    - :
      * factotum predefined rule
    - \*
      * sometimes an observation cannot be attributed to a specific thing (?), later it will belong to something that could be named [an entity?]
      * so Factotum will assign it an “arbitrary, unique name” to the observation 🡪 *indicated by the asterisk marker on the fact*
    - “
      * same subject as previous line
    - #
      * informal fact
    - –
      * continuation of fact (formal or informal)
    - #”
      * informal fact with same subject
    - #\*
      * informal fact with unique subject
    - #-
    - :<
      * (15) specifies inclusion of another file
      * files can include any other file, except a file which as already been included
      * the factotum configuration file in any installation provides a library path to search for included files [???]
    - :”
      * factotum predefined rule with same subject
    - blank
      * fact (only necessary if the users fact begins with
        + \* “ : # -
* **Subject**
  + The first word token on a line of a fact unless a marker is present
  + Ex:
    - SAT
    - Apollo
    - Virtual
    - B394
    - 123
* **Alias** 
  + Series of word tokens that provide an alternative name for an entity
  + Ex:
    - Vertex Cover
    - 3-Satisfiability
* **Predicate**
  + The objects’ names and context which occur after a subject and before the first citation
  + Ex:
    - Is the mother of C3PO
    - Cook’s theorem
* **Object** 
  + A series of word-tokens within a predicate that are terms in a matching vocabulary rule
  + Ex:
    - Zeus
    - Vertex Cover
    - Purple
* **Context** 
  + A series of word-tokens within a predicate that constitutes a series of terminal tokens in a matching vocabulary rule
  + Ex:
    - Is the father of
    - Beheaded
* **Citation**
  + A series of nested (?) bracketed references at the end of a fact
  + Ex:
    - [Uzgalis, 92]
    - [Chen 76]
* -(13)
* “Unlike natural language processing which tries to syntactically analyze some sentence into a tree structure with the parts os speech and their relationships recognized. Factotum is only interested in *entities and relations.”* 
  + “It only has to recognize subjects, attribute values [context?] and object names.”
  + **“*this means that syntactic analysis must only breaking up the predicate into object-names, attribute values, and relations”***
* **Observation** 
  + Record of an external event represented in computer symbols
    - Eg some baseball game stats
  + Can be associated with a citation that justifies the fact
    - Eg some baseball almanac
  + External event can be true, imaginary, literary or formal
  + ***when recorded with a citation in a FACTOTUM DATA FILE it is called a FACT***

(33)

“Factotum provides a way to generate new data that is a transformation of the existing data. This allows one to reorganize the data for the following purposes:”

* + *To communicate with other people* 
    - It is desirable to have a natural lang like factotum facts
  + *To build other models on the data*
    - Must put facts into the formats of other models, such as statistical models, knowledge representation models, etc
  + *To change the subject of concern for a relationship [???]*
    - It may be necessary to rewrite the fact to swap the positions of the subject and the objects and then reintegrate them into the data
  + *To distribute information and provide consistent cross reference material*
* -(9)
* “Data: recorded as simple text in a flexible lexical form called a fact” [not really flexible…]
* “The only constraint on the form of a fact is the first column of every line , after that blank is the only significant reserved symbol. Blank space is used to separate lexical entities [words??].”
  + “The first set of words is assumed to be the subject (or entity) name; the remainder of the line is attribute or relation [typo] to other entities.”
    - //what about a simple fact about itself, why always have to be related to others
* *“A collection of facts with the SAME SUBJECT is called an entity” //*is the name of the entity then the same as the name of the subject
* **FACT FILES DEFINE ENTITIES**
* see markers in General heading
* **Observation** 
  + Record of an external event represented in computer symbols
    - Eg some baseball game stats
  + Can be associated with a citation that justifies the fact
    - Eg some baseball almanac
  + External event can be true, imaginary, literary or formal
  + ***when recorded with a citation in a FACTOTUM DATA FILE it is called a FACT***
* **Formal fact: (14)**
  + Can be parsed successfully based on the specifications given by the user in the vocabulary associated with the fact
    - **WHAT DOES “parsed successfully” mean?**
  + (18)
* **Informal fact**
  + Ignored and carried around in the entity
  + May be incorporated as formal fact later,
    - Or
  + Just supplementary info about entity
  + (18)
  + Marked with #: Indicates there isn’t a formal description for this observation
  + *Sometimes observations don’t seem to fit the pattern of relations that seem to recur in the data—but observations still seems significant so include it*
  + **Informal facts attach themselves to the entity (or subject) that they occur in –** move around with the entity
* FACTS WITH NO MARKER ARE ASSUMED TO BE FORMAL (? WHY)
* **Note: comments** may be inserted in to facts with ‘\#’ inserted (comment ends at the end of the line)
* **4 SPECIAL KINDS OF PREDEFINED FACTS** 
  + indicated by : [why need colon if predefined, why need to include??]
  + symbolic rather than linguistic 🡪 LANGUAGE INDEPENDENT
    - ->
      * relation means that the SUBJECT is an ALIAS for the primary entity name mentioned in the predicate //wtf
        + :y-> x.

y has alternative name x

* + - <-
      * SUBJECT can be represented by the series of tokens following the arrow
      * Names and/or phrases become equivalent
        + :z <- x y.

z has the alternative multiword name x y

*//note, doesn’t explain next 2 predefined facts in text*

* + - < >
      * specifies the “type” the term is (note more than one type may be associated with a subject)
        + :y <x>

y is of type x

* + - :< or is it <= //PG 15 typo??? [note probably :< since predefined facts need colon]
      * (15) specifies inclusion of another file
      * files can include any other file, except a file which as already been included
      * the factotum configuration file in any installation provides a library path to search for included files [???]

EXAMPLES/OVERVIEW:

:primary term <- alias.

:primary term <- multiword alias.

:single-word-alias -> primary-term.

* + - :\* :< “file name”. //???

:primary term <type>.

* **ALIAS** 
  + Define an alias using a factotum predefined relation
  + Use alias in any predicated where reference to the primary name is needed
  + Improve readability of facts
  + **SUBJECTS** in facts must still be a SINGLE token
* **SUBJECT** 
  + When bringing in data into Factotum (aka recording a fact) *a subject for each observation needs to be determined*
  + Remember: subject is the first ‘word token’ in the line
  + Can be:
    - Name of a clearly defined ENTITY (already)
    - (most likely) subject of some literary, scientific, or fantastic observation [language, questionable?]
  + Note:
    - User may (prob) not understand what entities are being modeled [??--> AKA] 🡪 Still have observations, just don’t know subject ,
      * or sometimes no need to name subject
        + Ex: baseball scores or something
        + So mark subject with an \*, means that give it a unique identifier, but it’s not provided by user
    - **EXAMPLES OF NONLITERARY FORMS OF DATA: pg 16**

(16)

However the data is expressed, *must be a clear [? Umm] unique way to identify the relations in the data that can be expressed using the Factotum vocabulary 🡪 [expressed by user though??]*

* + requires unambiguous communication //lol wut? It’s pretty ambig
* **PREDICATES**
  + Predicates are the tokens after the ~~end of the~~ subject and before the end of the fact, or beginning of citations
  + Expresses:
    - attributes of a subject,
    - or relations between a subject and some other entity/entities
      * note: *to create a LINK between an entity[subject?] referenced in a predicate and the current subject, the token or phrase that represents the entity needs to be recognized [by the software duh]*
        + done by writing DESCRIPTIONS OF FACTS in a VOCABULARY which shows how a user wants to express his info

usually one vocabulary rule and many facts which fit that rule

* + domain of the researcher (only reserved symbol is the comment symbol /#)
* in writing facts: MISTAKES ARE INEVITABLE
  + inconsistent data should be identified, weeded out or made more precise
  + mistake exs:
    - spelling, mistyped names
    - reuse of name for something already been defined ??
    - inconsistent use of a relation, [aka, should apply to a different type of entity]
    - deeper errors, eg think recording a universal truth, but turns out to be inconsistent
      * (can resolve it by introducing additional qualifications to original data)
  + factotum tries to help user to create CLEAN FACTS by matching intent as recorded in the vocabulary against the individual facts entered
    - forces user to:
      * understand the essential distinctions,
      * similarities
      * relationships in the data and the facts
  + *factotum warns when fact ambiguous and be interpreted in more than one way*
* **CITATIONS** 
  + Specify source of the information in the predicate
  + [] at the end of a fact, can be nested
    - …killed. [ [KJB Gen 4:2] [KJV Lev 1:8] }
  + more than one citations for a fact, occur as a series within a containing pair of brackets
* **ENTITIES** 
  + All facts with the same subject are grouped together
  + THE SUBJECT IS THE NAME OF AN ENTITY (18)
  + First thing that F. processing does: collect together all facts with the same subject
    - *Merges duplicates of the entity name*
  + *🡪 the grouping of facts by entity is fundamental to the operation of most of the Factotum tools*
  + ssort ??
  + allows entities to be classified by **TYPE**
    - meaning and organization of types are defined by user
    - few primitive factotum types based on the lexer
* *“All factotum files consist of two, three, or four fields”* 
  + *OPTIONAL marker*
  + *Subject (generated or expressed)*
  + *Predicate*
  + *OPTIONAL citations* 
    - *🡪 [marker] subject predicate [citations]*
* SUBJECT is a single tag [???] terminated by the first whitespace char
* PREDICATE is user defined series of tokens
  + represent relation identification, attribute vals, or object/entity names [wait what]
* ***TOKENS***
* -sequence of chars, occur between separators (blank space or punctuation)
* Lexical analysis in Factotum is different from most lexical software:
  + Process split between
    - LEXICAL PROCESSING
      * Recognizes only primitive tokens
    - SYNTACTIC ANALYSIS
      * Primitive tokens may be combined to make a more complex token to match what is required
  + (19) “this allows user to have some control over the way tokens are recognized within his own facts without having to define his own lexical analysis”
  + *pg 19 – token chart /??*

//feels like a misnomer? Calling it vocabulary

* [My own obervations] (9, 21)
* Vocab file contains collection of rules, define relations
  + Rules = descriptive lines
    - simple text in a notation which allows the specification of the fact syntax and some semantics of relations.
  + In order to validate and manipulate the facts, need to have way to distinguish what parts of the predicate are:
    - An expression of a relation
    - What parts are references to other entities
  + User must define the syntax of what he wrote in terms that
    - SEPARATE predicate info from entity names
  + **Aka, vocab file is format for facts/relations**
* [from data section]
* **PREDICATES**
* Predicates are the tokens after the end of the subject and before the end of the fact, or beginning of citations
  + Expresses:
    - attributes of a subject,
    - or relations between a subject and some other entity/entities
      * note: *to create a LINK between an entity[subject?] referenced in a predicate and the current subject, the token or phrase that represents the entity needs to be recognized [by the software duh]*
      * **done by writing DESCRIPTIONS OF FACTS in a VOCABULARY which shows how a user wants to express his info**
      * **usually one vocabulary rule and many facts which fit that rule**

< >

* + represents an entity
  + vocabulary rules always start with this
    - because facts always start with a subject
    - the way in which a relation is represented in the vocabulary file constitutes the rest of the rule (??)
    - note: if Vocabulary rule contains additional <> markers, it represents relations to other entities
      * *eg* 
        + := “<> bears some relation to <>”
    - if it only has attributes,
      * it defines some properties of the entity
        + := “<> is purple”
  + note, since < and > are reserved symbols, if the user wishes to actually use these as characters, then they would be matched with \< or \>
* **fact rules** 
  + provides the syntactic pattern for a relation as it is represented in the data
  + the fact pattern always starts with the class of subjects allowed [???]
  + marked by predicate
    - **[name of relation] :=**
  + head of a language that can be generated from the fact rule by:
    - replacing the **nonterminals** (<nonterminal>) by its own definition,
    - and keeping other tokens, which are terminals, unchanged
      * **nonterminals** can be defined by phrase rules in the vocabulary, or by type classes from the symbol table
        + *note:* ***type classes*** *are determined from a pre-pass of the data,*

*matching is to that extent context sensitive*

* **phrase rules** 
  + a rule that describes some part of a fact and therefore does not start with a subject description
  + marked by the key words “phrase is”
    - or
    - **-=**

rules with the same ~~subject~~ name are alternative descriptions of the same relation

* + both fact and phrase rules may have alternatives
  + BUT: a fact rule cannot have a phrase rule alternative and vice versa
    - eg:
      * date -= “<day:n>-<month:n>-<year:n>”.
      * “ -= “<year:n>.<month:n>.<day:n>”.
* Factotum syntax rules are CFG
  + <nonterminal>
    - because these types can be specified for tokens, come context sensitivity is provided
  + terminal tokens are given as unmarked tokens [???]

a fact matches a vocab rule if

* + the fact is parsed by the vocab rule
  + if matches more than one rule…
    - **complains about ambiguity** 
      * ambiguity in the spec is not checked in the abstract, however if it occurs in practice the user is warned
* *Factotum vocab is capable of specifying the syntax of facts* 
  + \*\* syntax is aimed at:
    - formally separating the TOKENS that represent entities from the tokens that represent relationships
    - uses parser to match vocab rules to facts
* vocab defines the general form,
* and to a limited extent – the meaning of relations
* **goal of the parsing problem is to recognize ENTITIES and separate them from CONTEXT so RELATIONS between entities are recognized**
* note: a reference to an entity in the predicate can be a multiword alias
  + vocab should only define relations and types
    - 🡪 therefore alias and type information should be specified in FACT files using predefined facts
* just as the truth of facts is assumed/unimportant , the truth of relations is ignored
* **TYPES**
* (23)
* basically, in fact file, predefined fact, tells you type, and aliases
  + from this you can check if it matches the rule
* there can be many type trees

**Taxonomies**

* + factotum also contains constructs to help the user create one or more taxonomies of entities
    - *vocab may be used as an INTERFACE for writing rules to formally classify entities according to what* :
      * relations
      * attributes
      * types
      * and objects of relations
        + *all the objects embody*
        + 🡪 [vocab applies this] **to data and checks that an entity does not fall into more than one taxonomic category**
  + taxonomies can be used to generate strong types or attributes that are integrated back into data🡪 PROVIDES USER WITH MECHANISM FOR TYPE INHERITANCE
  + weak classification allowed too
    - **compound attributes** 
      * created using bools
      * compound attribute allows any logical combo of attributes to create a new attribute 🡪 which in turn may be used to create a new attribute 🡪 which may be used in entity constraints
      * attribute definition, and relation definition
  + note: attributes can be strings or numeric

3 methods of imposing semantic constraints:

* + **TYPE MATCHING (2.3.2)** 
    - Subjects and objects can be confined to predefined groups of entities (eg by using <> )
  + **RULE-RESTRICTIONS (p 26)** 
    - If-then-else construct that performs value checking on terms within a rule
    - Placed in the vocab after a fact or phrase rule
    - Applied in the order they are given to each fact or phrase that is parsed by the rule
    - After successfully parsing a fact or phrase by a vocab rule, the rule restrictions that follow the rule will be executed one by one
    - *Execution of rule restrictions does not affect the parsing of rules* 
      * Ex:
        + If- then

?< *condition*

: *command* >?

* + - * + If-then-else

?< *condition*

: *command*

: *command* >?

* + - * + If-then-elseif- then- else //note, may repeat then-elseif as mnay times as you’d like before then-else

?< *condition*

: *command*

?: *condition*

: *command*

: *command* >?

* + **ENTITY CONSTRAINTS**
* CONDITION
  + Logical expression that returns a truth value
  + Arithmetic ops on numbers
  + Comparison ops on numbers and strings
  + Logical operations on sub-expressions for condition evaluation
  + *To reference a term matched by a non-terminal, the nonterminal should carry a label* 
    - Labels are optional, a word token
  + **Toke type specification**
    - Allows a list letter keys with optional preceding dash
      * If dash is present: following types of chars are NOT permitted in the token
    - No specification means no restriction on the type of tokens matched

COMMANDS

* + If the condition is true, the command in the “then” part will be executed. Otherwise, the command in the “else” or “else-if” pare will be executed
    - Satisfied
      * Satisfied with this check and proceed to next check
    - Comment “*message”*
      * Give a comment and do next check
    - Warn “*message”* 
      * Give a warning and do the next check
    - Error “*message”*
      * Give an error message and by-pass all other checks for the vocabulary rule
    - Abort “*message”* 
      * Give an error message and abort the program
    - Skip N
      * Skip the next N checks from the current check

SPECIFYING RESTRICTIONS ON ENTITIES [ **ENTITY CONSTRAINTS]**

* + a Factotum entity is a collection of facts with the same subject
  + there are conditions and commands for entity restrictions 🡪 same as for predicate objects
    - CONDITION in a fact restriction refers to non-terminals in particular fact rule
      * Condition in an entity refers to things about the entity
      * Specifying the name of a relation, preceded by back slash yields a count of the number of times that relation occurs in this entity
        + \$ -- refers to name of current entity

\$f 🡪 means the father relation, which is labeled by f (p 29)

note: \$f is the same as \$f.o [if relation is unary]

* + - * + \$tag:label

entity name referred to by the object in relation tag with label

* + - * + \& -- refer to number of facts in entity (including factotum special facts)
        + \@ -- refer to the number of diff relations in the entity (excluding special facts)
        + \\* [relation tag] – refer to the number of those relations in the entity
        + \tag

true if tag relation exists in the entity

* + - * + \\*<>tag:label

number of type(s) of the tag label in entity

* + - * + \<>tag: label

type(s) of the tag label in the entity

* + **entity constraint includes a header and a collection of if-then-else constructs** 
    - the subject specifies the type of entity to be checked
    - \*\*\*this is how we use these symbols 🡪 inside the condition blocks!!!
* **Labeling objects, p 27**
* **Implications and Abstract Models:**
  + Eg a relation can be implied by another relation 🡪 doesn’t need to be made explicitly by factotum, can be implied by a vocab rule
    - ABSTRACT SEMANTIC MODEL
      * Factotum vocab provides interface for the ASM’s,
      * Each relation in vocab may contain rules that link the objects in a relation to the abstract model
* Note: since appearance of facts already restricted by the vocab 🡪 if want to change one, must change the other “**MODEL REORGANIZATION”** 
  + Most of the time, more facts than number of vocab rules
    - more practical to specify changes in vocab and let a tool do changes automatically
    - **Generate Rule**
      * Always inserted after a vocab rule
      * String object in the generate rule specifies how to reorganize that fact
      * Non-terminals in a generate rule must have been defined in the corresponding vocab rule and are referenced by their labels
      * No limit on the context or how many non-terms in original vocab rule are used in the gnerate rule
      * Possible to have more than one generate rules
      * Ex p 34 \* (good example)

-is a “subject” always the first item in a fact? Can it occur as an object as well? Can objects be subjects? (12)

-why do we need aliases? Are they necessary? Why would we want an alternative name for an entity? Is this different than a subject ? object? (12)

-how are citations considered “nested”? in that they are within the fact between predicates etc or nested within eachother?

~~-fact == observation ? [in a word yes]~~

-what’s a predefined factotum rule? ~~Informal fact ?~~

* - WHAT DOES “parsed successfully” mean? [in formal facts] pg 14
* - keep interchanging fact and entity WHATS UP WITH THAT
* -why are facts with no marker assumed to be formal
* -subject == primary term?
* -:< or is it <= //PG 15 typo???
* (15)
* -um what is the sentence saying?
  + “Subjects in facts must still be a single token to allow manipulating data before the syntactic model has grown up enough to specify the entity names and types.”
* (9)
* -“The first set of words is assumed to be the subject (or entity) name; the remainder of the line is attribute or relation [typo] to other entities.”
* //what about a simple fact about itself, why always have to be related to others
* -*“A collection of facts with the SAME SUBJECT is called an entity” //*is the name of the entity then the same as the name of the subject? What about aliases for the subject? Or then are they simply aliases for the entity?
* (18)
* “It is usually obvious from the error messages from checking the facts what has happened and qualifications to the names need to be created” ???
* -Last paragraph in 2.3 , p 18 – what does that even mean??
  + “the only type constructor is multiple definitions of a type that specify other types that are members of the type being defined”
  + “this allows multiple type trees to be constructed”
* (19)

So by tokens he means chars? Cuz tokens are usually words etc

(22)

“Rules with the same subject are alternative descriptions of the same relation”

* + does he mean rules with the same NAME – since the subject is always the first thing
* (23) figure 12 is a confusing/bad example
* (27) objects == entities?
* (35) says that “markers, subjects, predicate, and citation are lexically recognized for all facts” 🡪 if vocab rules count as facts, what’s the subject??
  + Subject is the name of the rule \*\*\*