Errata and Comments on "The Logic of Typed Feature Structures"

p. 5 18th line: 'turn quasi-linear unification algorithms into quadratic ones' should be 'quasi-linear ones with a heavy constant overhead'

p. 11 3rd line: 'a feature structure is modeled by a possibly cyclic directed graph' should be 'a feature structure *is* a possibly cyclic directed graph'

p. 20, theorem 2.5 'pre-order' should be 'partial order'

p. 21 Theorem 2.6: 'unique' should 'unique up to isomorphism'

p. 21 5th line of proof: 'or if' should be 'and'

p. 23 5th line: 'This results in a system in which every possible disjunction of species information is represented by a type. Of course there are 2^n types derived from King's construction if there are n species, which is a worst-case scenario for the upper powerdomain construction.' "worst-case" here presumabably refers to worst for the purpose of an implementation. But this presumes a naive implementation of King's logic in which all 2^n types actually need to be explicitly represented.

p. 24 Definition 2.7: 'symmetric' should be 'symmetric and anti-reflexive'

p. 30 (13) add 'arg' to set

p. 31 (19) add to set {3rd,sing,masc} {3rd,sing,fem} & {3rd,sing,neut}

p. 31 first bullet: 'Dep(p,q)' should be 'Dep(q,p)'

p. 33 'Both PATR-II and the Rounds and Kasper systems treated feature structures intensionally, but enforced extensional identity conditions on atoms'. Shieber (1986) doesn't actually say this. I guess you have to refer to Pereira & Shieber (1984). One is left wondering where this idea came from since it is clearly inconsistent with the Shieber's (1986) type/token distinction.

p. 35 5th line 'Thus it is possible to model what is known about an object at some intermediate computation stage ...'. Why should we say 'model' here and not 'describe'. Afterall there is presumably some total linguistic object out there which we are collecting information about.

p. 45 second line under heading 'Unification': 'informational content' should be 'structure' (Defintion 3.4 never refers to informational content)

p. 46 Defintion 3.11 should be:

Definition 3.11

Suppose F and F' are feature structures such that

$$F \sim \langle Q, \overline{q}, \theta, \delta \rangle,$$
$$F' \sim \langle Q', \overline{q}', \theta', \delta' \rangle, \text{ and}$$
$$Q \cap Q' = \emptyset.$$

Let $\bowtie$ be the smallest equivalence relation such that

$$\overline{q} \bowtie \overline{q}', \text{ and}$$
$$\text{if } q \bowtie q',\ (\delta \cup \delta')(f, q) = \tilde{q} \text{ and } (\delta \cup \delta')(f, q') = \tilde{q}' \text{ then } \tilde{q} \bowtie \tilde{q}'.$$

$F \sqcup F'$ is defined iff for each $x \in (Q \cup Q')/_{\bowtie}$, $\{(\theta \cup \theta')(q) \mid q \in x\}$ has an upper bound.

If $F \sqcup F'$ is defined then $F \sqcup F' = \langle (Q \cup Q')/_{\bowtie}, [\overline{q}]_{\bowtie}, \theta^{\bowtie}, \delta^{\bowtie} \rangle$, where

$$\theta^{\bowtie}(x) = \bigsqcup\{(\theta \cup \theta')(q) \mid q \in x\},$$
$$\delta^{\bowtie}(f, x) \text{ is defined iff for some } q \in x,\ (\delta \cup \delta')(f, q) \text{ is defined,}$$
$$\text{if } \delta^{\bowtie}(f, x) \text{ is defined then } \delta^{\bowtie}(f, x) = \bigcup\{[(\delta \cup \delta')(f, q)]_{\bowtie} \mid q \in x\}.$$

p. 47 Add corollary to Lemma 3.12:

Corollary

For each $x \in (Q \cup Q')/_{\bowtie}$,

$$\text{if } q \in x \text{ and } (\delta \cup \delta')(f, q) \text{ is defined then } \delta^{\bowtie}(f, x) = [(\delta \cup \delta')(f, q)]_{\bowtie},$$

Proof

$$q \in x \text{ and } (\delta \cup \delta')(f, q) \text{ is defined}$$
$$\Longrightarrow [(\delta \cup \delta')(f, q)]_{\bowtie} \subseteq \delta^{\bowtie}(f, x)$$
$$\Longrightarrow [(\delta \cup \delta')(f, q)]_{\bowtie} = \delta^{\bowtie}(f, x).$$

since $[(\delta \cup \delta')(f, q)]_{\bowtie}$ and $\delta^{\bowtie}(f, x)$ are both equivalence classes of $\bowtie$

$\square$

p. 47 Theorem 3.13:f

Theorem 3.13 (Unification)

If F and F' have an upper bound then $F \sqcup F'$ is defined and $F \sqcup F'$ is a least upper bound[1] of F and F' in $\langle \mathcal{F}, \sqsubseteq \rangle$.

Proof

5th line '$\delta^* : (Q \bigcup Q') \times \mathsf{Feat} \to (Q \bigcup Q')$'. To be consistent with the rest of the text, this should be 'δ^*:$\mathsf{Feat} \times (Q \bigcup Q') \to (Q \bigcup Q')$'. The usage here, however,

[1] Carpenter has '*the* least upper bound'. But the concrete feature structures form a pre-order, not a partial order. It's not clear how the notion of least upper bound in a pre-order could be defined.

is consistent with usage in the rest of the automata theory and feature structure literature.

The proof is fine until the line

> "Note that $\bowtie$ is the minimal equivalence relation satisfying these two conditions, so that $q \bowtie q'$ only if $q \equiv_{h^*} q'$."

Certainly, "$q \bowtie q'$ only if $q \equiv_{h^*} q'$", but the proof is not entirely clear. The following seems clearer and is almost the "dual" of the original argument.

$\equiv_{h^*}$ is an equivalence relation, $\overline{q} \equiv_{h^*} \overline{q}'$ and $\delta^* = \delta \cup \delta'$.

Thus, by condition 1 of $\equiv_{h^*}$, equivalence relation $\equiv_{h^*}$ meets both conditions for $\bowtie$.

Thus, since $\bowtie$ is the smallest equivalence relation meeting these conditions, if $q \bowtie q'$ then $q \equiv_{h^*} q'$

In other words, "$q \bowtie q'$ only if $q \equiv_{h^*} q'$".

The proof is, by and large, fine again until the second to last line. This should read something like,

$\delta^{\bowtie}(f, x)$ is defined
$\Longrightarrow$ for some $q \in x$, $(\delta \cup \delta')(f, q)$ is defined
$\Longrightarrow$ for some $q \in x$, $\delta^{\bowtie}(f, x) = [(\delta \cup \delta')(f, q)]_{\bowtie}$ by the Corollary to Lemma 3.12
$\Longrightarrow$ for some $q \in x$,

$$
\begin{aligned}
& j(\delta^{\bowtie}(f, x)) \\
&= j([(\delta \cup \delta')(f, q)]_{\bowtie}) \\
&= h^*((\delta \cup \delta')(f, q)) \\
&= (h \cup h')((\delta \cup \delta')(f, q)) \\
&= \delta''(f, (h \cup h')(q)) \\
&= \delta''(f, h^*(q)) \\
&= \delta''(f, j([q]_{\bowtie})) \\
&= \delta''(f, j(x))
\end{aligned}
$$

$\Longrightarrow j(\delta^{\bowtie}(f, x)) = \delta''(f, j(x))$.

□

p. 48 4th line from the bottom. 'which is the most general feature structure ...'. The definite article shouldn't be used here since there isn't a unique most general feature structure in the pre-order of concrete feature structures.

p. 51 lines 15-16: 'partial objects' should be 'partial information about objects'???

p. 51 9th line from the bottom: 'described' should be 'modeled'

p. 53, third bullet from bottom: 'if F@π ... ' should be 'if F@π is defined and

F@π ...

p. 53 4th bullet: if $\delta(\pi_1, \bar{q})$ and $\delta(\pi_2, \bar{q})$ are defined and

p. 54 figure 4.1: 'PRED:(REL:sees $\wedge$ OBJ:john)' should be 'PRED:REL:sees $\wedge$ OBJ:john'

p. 54 first line: The satisfaction conditions in *4.2* are ...

p. 55 Theorem 4.4: This definition of 'monotonicity' does not correspond to th e normal logical definition:

Definition (Monotonicity)

Where T, T' are theories, if $T \models \phi$ and $T \subseteq T'$ t hen $T' \models \phi$

(Note that in this standard definition $\models$ means "entails", whereas in Carpenter's definition $\models$ means "satisfies".)

p. 56 5th line: 'The logics studied by Johnson, Smolka, and King are not monotonic in the sense of Theorem 4.4, due to their classical trreatments of description negation and implication.' Theorem 4.4 does not even apply to King's logic since '$F \sqsubset F'$ would have to be interpreted as subsumption between objects, which does not make sense. The whole problem arises as a result of the decision to consider feature structures as models of partial information. The alternative of considering feature structures as normal form descriptions is mentioned in the book but not really fully considered.

p. 56 8th line: 'satisfies' should be 'does not satisfy'

p. 56 Theorem 4.5: Insert 'MGSat(ϕ) is defined iff ϕ is satisfiable

p. 56 4th line from bottom: 'consistent' should be 'satisfiable'

p. 56 last two lines: delete 'by the number ... same number of features'

p. 57 1st line: 'conjunctions' should be 'colons and conjunctions'

p. 57 4th line: 'inconsistent' should be 'unsatisfiable'

p. 57 Delete 'we consider the case of cycles ... we take a similar move in taking'

p. 57, 3rd line from bottom to bottom: if MGSat(ϕ) $\sqcup$ MGSat(ψ) ... also satisfies ϕ ' should be 'MGSat($\phi \wedge \psi$) esists and F $\models \phi \wedge \psi$ iff MGSat(ϕ,ψ) $\sqsubset$ F$'$.

p. 58 8th line from bottom: 'since II is finite' should be 'since Hooks(F) is finite'

p. 59 The inductive step of the proof only works if the formula is already in disjunctive normal form.

p. 59 Second display in proof: Min is not well defined since the concrete feature structures form a pre-order

p. 64 14th line: 'The first axiom' should be 'The second axiom'

p. 64 6th line from the bottom: 'axioms' should be 'axioms and deduction schemas'

p. 65 3rd paragraph: 'the sort axioms' should be 'the type axioms'

p. 65 3rd paragraph, 3rd line: 'more specific than' should be 'at least as specific as'

p. 69 3rd line: 'Of course these two approaches are not really distinct. One way to think of the most general satisfiers is as a particularly compact way of representing the disjuncts of a normal form description.' Surely this is circular. If feature structures are just normal forms for descriptions, then descriptions are describing descriptions. Aren't there some empirical objects out there that you really want to describe?

p. 69: In the proof of Theorem 4.18. 'there must be a disjunct of $DNF(\phi)$ which is more general than some disjunct of $DNF(\psi)$'. Why is this the case. Maybe there's a disjunct of one that is simply incomparable with any disjunct of the other.

p. 74: 'Johnson showed ... in his system' Note King also shows this in his system.

p. 77 10th line: 'typing' should be 'appropriateness'

p. 80: 15th line from the bottom 'instead of using unordered lists of feature-value pairs as is usual in implementations of feature structures ... ' surely it's usual to keep the feature-value pairs in lexicographic order, isn't it?.

p. 83 2nd paragraph, 3rd line: 'feature structures into ...' should be 'objects into ...' (King does not consider featuer structures as objects)

p. 84 18th line from bottom: 'partitions the carrier set ... into overlapping sorts' this is a contradiction

p. 88 7th line above Well-Typed Feature Structures: 'To ensure that there is such a most general type at which every feature is introduced, the dual of inheritance could be performed so that features were taken to be appropriate for the meet of all the types for which they are appropriate, with a value restriction determined by taking the meet of all the restricitons on their subtypes.' This obviously won't work. Consider:

```
list
   list-sign
      ne-list-sign first:sign rest:list-sign
      e-list
   list-phon
      ne-list-phon first:phon rest:list-phon
      e-list % multiple inheritance
```

The meet of `ne-list-sign` and `ne-list-phon` is `list`, so `list` would have to introduce the features `first` and `rest`. By upward closure, then, `first` and `rest` would also have to be allowed for `e-list`.

p. 89 Theorem 6.3 after 'such that F $\in$ F$'$ ' insert 'TypInf(F) is defined iff F is typable, and '

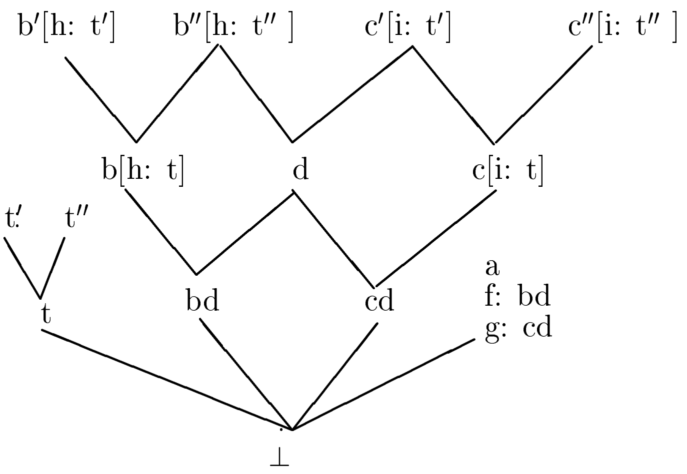
p. 90 Theorem 6.5: first line of proof: 'TypInf(F)' should be 'TypInf(F$'$)'

p. 92 Theorem 6.9 The theorem is correct but misleading. We don't need to use TypInf here. Only TypRan. The same comment holds for Corollary 6.10, Theorem 6.21 and Corollary 6.22.

p. 93 Corollary 6.10 This corollary leads one to believe that lazy type inference is a good thing for an implementation. One might mention that lazy type inference is not used in ALE.

p. 95 9th line: 'One instance of this is the HPSG grammar formaliasm, in which a well-formed sign is totally well-typed.' This leads one to believe that *total* well-typing is needed for an HPSG implementation, which is simply false. Even if the objects in HPSG are total, a parser will still work with partial descriptions of these obects. The real motivation for *total* well-typing is the positional encoding of features.

Also note with regard to total well-typing. Carpenter shows (p. 92) that we cannot conclude that $F \sqcup F'$ is typable just because $F, F' \in \mathcal{F}$ are typable. One might expect total well typing to help us in this case. But, in fact, even if F and F' are totally well typed, we still can't conclude that $F \sqcup F'$ is typapable. To see this consider the following signature:[2]



Then unify the following:

[2] The appropriateness conditions here are "continuous but not Join-Preserving". See Corollary 6.30 and Figure 6.7.

$$\begin{bmatrix} \text{a} & \\ \text{f} & \begin{bmatrix} \text{b} & \\ \text{h} & \boxed{1}\ \text{t} \end{bmatrix} \\ \text{g} & \begin{bmatrix} \text{c} & \\ \text{i} & \boxed{1} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{a} & \\ \text{f} & \text{d} \\ \text{g} & \text{d} \end{bmatrix}$$

Total well-typing does help with type inferencing, however, when the appropriateness conditions meet the join preservation condition (def. 6.29). In fact, if the appropriateness condition meets join preservation and F and F' are *totally* well-typed, then $F \sqcup F'$ is not just typable, it is in fact already well-typed. This might be mentioned as a second argument supporting the utility of *total* well-typing.

p. 96 7th line above Def. 6.14: 'it is a filter, because meets preserve well-typedness' To be a filter, the meets need to preserve *total* well-typedness.

p. 96 Definition 6.14 'there is a lest well typed ... ' should be 'there is a least *totally* well typed '

p. 97 Second bullet of (14). '$Approp(\sigma, \tau)$' should be '$Approp(f, \sigma) = \tau$ for some $f \in$ FEAT

p. 100 Line below proof of 6.19: 'The total type inference theorem' should be 'the total typability theorem'

p. 109 4th line from bottom: 'This results in a blend of syntactic and semantic representations of information; positive information is represented semantically in terms of feature structures, whereas negative information is maintained as a syntactic description.' This does not hold if a feature structure is held to be simply a normal form for a description. Then all of the information is syntactic.

p. 112 Second line after definition: 'equivalent' should be 'token identical'

p. 113 second bullet from bottom: after both instances of 'if $q \in Q$ insert 'and $\delta(f, q)$ is defined'

p. 114 4th line of proof: 'the proof the unificatin' should be 'the proof *of* the unification'

p. 114 (9): insert after 'if and only if' $\delta(\pi_1, \bar{q})$ and $\delta(\pi_2, \bar{q})$ are defined

p. 117 'Fully Inequated Feature Structures': The fully inequated feature structures seem to be like the totally well typed feature structures in that a lot of redundant information is added into the feature structure. Normally adding redundant information into feature structures conflicts with the idea that feature structures are intended to be used computationally. For totally well typed feature structures, one can make the case that the advantage of positional encoding counterbalances the disadvantages. But what is the motivation for fully inequating?

p. 121 Theorem 7.14 '$FIneq : \mathcal{IF} \to \mathcal{IF}$' should be '$FIneq : \mathcal{IF} \to \mathcal{FIF}$'

p. 124 4th line in Identity in Feature Structures: 'feature structures' should (perhaps) be 'complex feature structures'

p. 124 5th line in Identity in Feature Structures: 'The standard models of these structures employ directed graphs ...' What does he mean here? These structures *are* directed graphs. See comment to page 11.

p. 124 bottom: 'Smolka (1988) partitioned the atoms according to whether they were copyable or not.' The distinction is really a distinction between constants and predicates.

p. 125 7th line: 'F and G' should be 'SUBJ and OBJ'

p. 128 Fig 8.2: ARG1 and ARG2 are reversed in the FS located at <ARG1>

p. 129 second line beloc (4): 'strongly' should be 'totally'

p. 130 second line: 'Prolog II and Prolog II' should be 'Prolog II and Prolog III'

p. 130 last sentence, first paragraph: 'used provide' should be 'used to provide'

p. 131 in the proof: 'nodes assigned a common ... ' should be 'nodes assigned types with a common ... ' and 'all appropriate feature values' should be 'all appropriate feature values for that supertype'

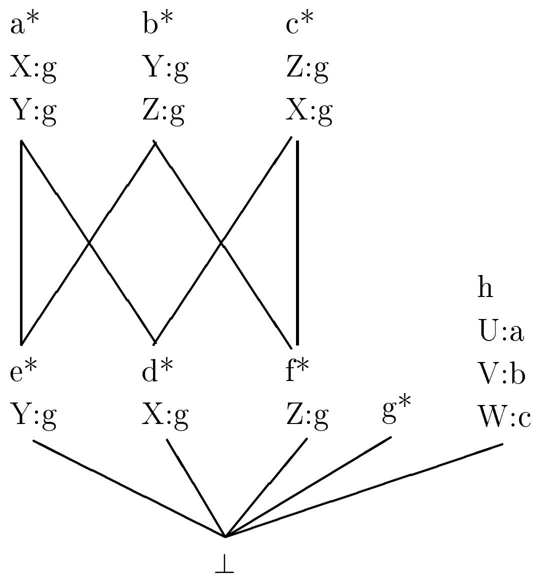
p. 133 middle of page: 'to assume that only extensional types are maximal' must be 'only maximal types are extensional'

p. 136 Feature structure (10) is inconsistent. $\perp$ allows features h and j but σ which is subsumed by $\perp$ allows only feature f.

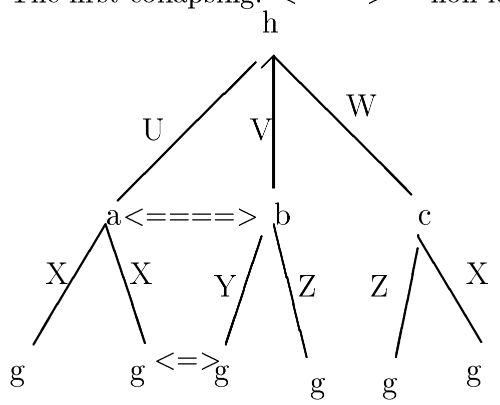
p. 136 sentence above theorem 8.11: 'The first thing to note is that we have indeed strengthened our extensionality conditions.' Theorem 8.11 does not make this point. We also need to note the the feature structures in (9) and (10) are extensional but not collapsed.

p. 137 first sentence of the proof: 'First note that the transitive, reflexive closure of the union of two collapsings is also a collapsing.' This is not true, as can be seen from the following example:

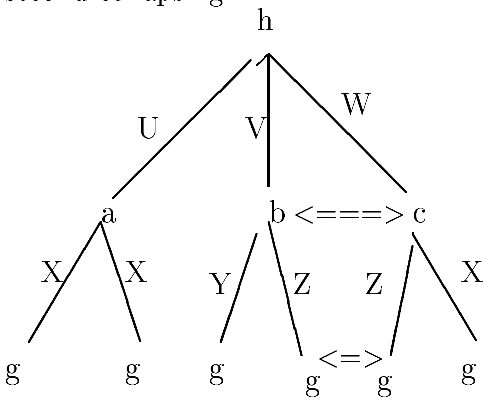
```
Type hierarchy:
---------------

Lower case = type
Asterisk = extensional type
Upper case = attribute
```



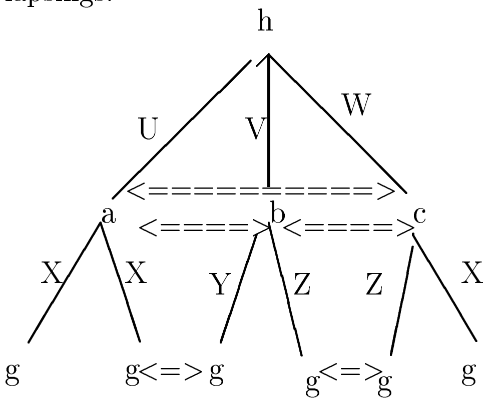
The first collapsing: <===> = non-identical equivalent nodes



second collapsing:



The transitive, symmetric and reflexive closure of the first and second collapsings:



This isn't a collapsing.

p. 139 15th line from bottom: 'a term subsumes a second term' should be 'a term properly subsumes a second term'

p. 142 after (2): 'Unfortunately, F and F' are not alphabetic variants because although $F \sqsubseteq F'$, we do not have $F' \sqsubseteq F$.' Perhaps what one needs is a different definition of $\sqsubseteq$. It's not clear that def 3.4 captures the intuitive, but never defined, notion of "information content".

p. 161 8th line from the bottom: 'This means that the description of such a feature structure is only satisfiable if we consider models with non-maximal types' It's clear that this is not a model for HPSG, where objects are considered to be total.

p. 164 23rd line 'Smolka (1989)' should be either 'Smolka (1989a)' or 'Smolka (1989b)'.

p. 165 third line '$F'_{h(q)}$' should be '$F'_{h_Q(q)}$'

p. 169 second line form bottom. 'can be expressed as the limit ...' should be 'can be expressed as the least upper bound ....'

p. 170 12th line from the bottom: 'Note that even for countably based domains, there may be a countably infinite number of domain objects.' Shouldn't this be an 'uncountably infinite number'?

p. 170 last two lines: 'Furthermore, an element of D is compact if and only if it is not the limit of any infinite ω-chain'. From the definition of ω-chain, one would think that such a chain is necessarily infinite. Also, the last two lines of the proof: 'Thus the chain X = $\{x_i, \ldots, x_n\}$ is finite.' The notion of a chain was never defined. Only ω-chain, which shouldn't have a final member, x_n.

p. 170 4th line: $y_n = \bigsqcup_{m<n} x_n$' should be $y_n = \bigsqcup_{m<n} x\ _m$.'

p. 174 last line of proof: '$Id(x \bigsqcup y) = Id(x \bigsqcup y)$' should be '$Id(x \bigsqcup y) = Id(x) \bigsqcup Id(y)$'

p. 175: First line of proof 'Define a function $\phi : Comp(D) \rightarrow D$' should be a function '$\phi : Ideals(Comp(D)) \rightarrow D$'

p. 175 last 3 lines: 'Thus a set $\{F_1, \ldots, F_n\} \subset \mathcal{F}$ of feature structures is taken to describe objects which are at least as specific as one of the F_i.' Now feature structures are taken to be descriptions again, and were assuming a specificity ordering on objects?!

p. 175 17th line from bottom. 'It is more often than not the case that there are F'' such that $F \sqcap F' \sqsubseteq F''$, but $F \not\sqsubseteq F''$ and $F' \not\sqsubseteq F''$. In the chapter on the type hierarchy, we saw that just this kind of behavior caused distributivity to fail ....' This kind of behavior also occurs in distributive lattices.

p. 178 Corollary 12.16, the restatement of "our representation theorem as teh following corollary to our representation theorem:" '... is order isomorphic to ...'. Shouldn't this be a plain isomorphism rather than an 'order' isomorphism. There was no mention of an ordering on descriptions.

p. 180 4th & 5th lines: '... entails that X must also be bounded. Thus we have $f(\bigsqcup X) = \bigsqcup(f(X)) = \bigsqcup Y$ by the continuity of f.' This result follows if X is directed, not necessarily if X is merely bounded.

p. 182 4th line: 'For instance, just because $Fill$ is a closure operator does not mean the $Fill(F \bigsqcup F') = Fill(F) \bigsqcup Fill(F')$, because $\{F, F'\}$ might not be directed.' It ought to be sufficient for $\{F, F'\}$ to be bounded since domains are bounded complete. The first line of the proof of theorem 12.21 also says 'We need to show that $f(D)$, the image of the closure, is closed under **bounded** joins.' (emphasis mine)

p. 186 13th line: 'the PATR-II system ... allows context free grammar rules ... ' Shieber shows that PATR II doesn't require a context free back bone. Rules like VP $->$ VP X are allowed.

p. 188 last two lines: 'we exploit the duality between feature structures and (normal form) attribute-value descriptions'. Why not just say feature structures *are* normal form attribute-value descriptions? After all, you need some logical form that you can use computationally. That's the purpose of feature structures.

p. 189 16th line: 'if an expression **a** is assigned to a category F and $F \sqsubseteq F'$, then the expression **a** is also assigned to the category F'.' This seems like a substantive claim, but then in the next sentence we see that it simply falls out from the definition of "assigned to". The natural question is: If `john` is a word, and word $\sqsubseteq$ verb, does that mean the `john` is a verb? The answer is apparently, that `john` would not, by definition, be *assigned to* the category `word`.

p. 208 Defintion 13.14: 'most general morphism' not defined

p. 227 Middle of the page: 'The axioms in (1) are weak enough that type inference can be performed in linear time.' If this is true of cyclic structures, then I would like to know how to do that.

p. 229 Definition 15.2: '$\mathsf{Cons}^*(\sigma) = \bigwedge_{\tau \sqsubseteq \sigma} \mathsf{Cons}(\sigma)$' should be $\mathsf{Cons}^*(\sigma) = \bigwedge_{\tau \sqsubseteq \sigma} \mathsf{Cons}(\tau)$

p. 233 Proof of theorem 15.9: 'This is because π-resolution is obviously monotonic in the sense that if $G \sqsubseteq G'$ and $G \stackrel{\pi}{\Rightarrow} J$, then there is a J' such that $G' \stackrel{\pi}{\Rightarrow} J'$ and $J \sqsubseteq J'$.' This is false. Consider the following example:

```
bot
   a f:d
      b f:d
      c f:d
   d
     d1
     d2

Constraints: a => f:d, b => f:d1

G =
a
  f d

G' =
b
  f d2
```

We have $G \overset{\xi}{\Rightarrow} G$, so $J = G$. But what about $G' \overset{\xi}{\Rightarrow}$?? This can not be resolved. And even if Carpenter's assumption were true, the next line still wouldn't follow (imho). I think the theorem's true anyway, but the proof...

p. 238 Theorem 15.12: '$AbsVal(Length(\pi) - Length(\pi')) \leq k$' should be $AbsVal(Length(\pi) - Length(\pi'')) \leq k$, if I've understood this right.

p. 242 Definition 15.13 quote from Pereira: "There seems to be something wrong in Definition 15.13. I believe G@π should be G_π, where G_π is the result of resolving F along path π"

p. 244 just below '(9) $F = F_i$' should be '$F =$ 'F_0'