ADRIAN BRASOVEANU[*,†] (abrsvn@eden.rutgers.edu)
ALAN PRINCE[*] (prince@ruccs.rutgers.edu)

RANKING ARGUMENTS AND THE FUSIONAL REDUCTION ALGORITHM

1. THE PROBLEM AND THE PROPOSAL.

(1) Evidence for a particular constraint ranking (i.e. the 'primary' set of data):
  (i) the **violation profile** of a set of candidates (where a candidate is an I/O pair);
  (ii) the designation of a particular candidate as **the desired optimum**.

Tableau 1: the violation profile of **three** candidates (**a**, **b**, **c**) with respect to **three** constraints (**1**, **2**, **3**). The number in a cell: the number of 'offending' structures of a particular candidate with respect to a particular constraint.

Tableau 1	**1**	**2**	**3**
a: <I, O_a>	1	1	1
b: <I, O_b>	2	0	2
c: <I, O_c>	1	2	0

Desired optimum: **a: <I, O_a>**

A first formulation of the problem:

(2) what are the necessary and sufficient ranking conditions enforced by the 'primary' set of data, i.e. what are the constraint rankings that satisfy it?

Based on **Tableau 1**, we generate the comparative **Tableau 2** below, containing **two** candidate comparisons: **a** (optimum) ~ **b** (suboptimum); **a** (optimum) ~ **c** (suboptimum)

Tableau 2	**1**	**2**	**3**
r_1: a~b	W	L	W
r_2: a~c	e	W	L

- **W: the constraint prefers the desired optimum (the Winner)**
  |optimum violations|<|suboptimum violations|
- **L: the constraint prefers the suboptimum (the Loser)**
  |optimum violations|>|suboptimum violations|
- **e**: **the constraint does not distinguish between the compared candidates**
  |desired optimum violations|=|suboptimum violations|

A comparative tableau encodes the requirements that the 'primary' set of data enforces on the constraint ranking of the language under consideration:

*(3) A constraint ranking **satisfies** a comparative tableau **iff***
for each tableau row and constraint that assesses an L,
there is some constraint that: (a) assesses a W in that same row;
(b) dominates the L-assessing constraint.

e.g. **Tableau 2** is satisfied only by **1>>2>>3**:

---

[*] Rutgers University.
[†] University of Frankfurt.

r_2 requires **2>>3**.
r_1 requires 1>>2 OR 3>>2.
But since 2>>3 holds (by r_2), 3>>2 is false; hence, we infer from r_1 and r_2 that **1>>2**.

Given that **a comparative tableau is a (partial) characterization of the constraint ranking of a particular language…**

THE PROBLEM:

(4) what are the necessary and sufficient ranking conditions enforced by a tableau, i.e. what are the constraint rankings that satisfy it?

(5) **Tableau 2** does not provide a perspicuous way to answer this question:
- the ranking of constraints 1 and 2 is not explicitly displayed;
- the ranking of constraints 1 and 3 is not explicitly displayed.

Tableau 2	**1**	**2**	**3**
r_1: a~b	W	L	W
r_2: a~c	e	W	L

(6) **Tableau 3** goes halfway towards this goal: 1>>2 is explicitly displayed (r_3).
Tableau 4 gives a complete answer: 1>>2 and 1>>3 are explicitly displayed (r_4).

Tableau 3	**1**	**2**	**3**
r_3: d~e	W	L	e
r_2: a~c	e	W	L

Tableau 4	**1**	**2**	**3**
r_4: f~g	W	L	L
r_2: a~c	e	W	L

(7) **Ranking conditions and comparative tableaux:**
  (a) **W**'s introduce **disjunctions**, e.g. r_1=WLW requires 1>>2 OR 3>>2
  (b) **L**'s introduce **conjunctions**, e.g. r_4=WLL requires 1>>2 AND 1>>3

(8) The problem becomes more pressing when the number of constraints and the number of comparisons under consideration is larger – see Chickasaw (pp. 13-14).

THE PROPOSED SOLUTION:

(9) For each comparative tableau, there is a unique canonical comparative tableau called **Maximally Informative Basis (MIB)** that:
  (a) is **equivalent** to it;
  (b) contains a **minimal number** of **independent** rows;
  (c) displays in a **perspicuous / maximally informative** way all the necessary and sufficient ranking conditions.

(10) We give **an algorithm** called **Fusional reduction (FRed)** that:
  (a) outputs the **MIB** for any input tableau;
  (b) can identify if the input tableau is contradictory and the minimal set of rows in the input tableau that yields a contradiction;
  (c) can perform Recursive Constraint Demotion (**RCD**).

Example: see Chickasaw (pp.13-14).

2. THE FUSIONAL REDUCTION ALGORITHM FOR OBTAINING THE MIB.

In this section, we will incrementally construct and motivate the steps of the algorithm.

Consider again **Tableaux 2**, **3** and **4** below.

Tableau 2	**1**	**2**	**3**
r_1: a~b	W	L	W
r_2: a~c	e	W	L

Tableau 3	**1**	**2**	**3**
r_3: d~e	W	L	e
r_2: a~c	e	W	L

Tableau 4	**1**	**2**	**3**
r_4: f~g	W	L	L
r_2: a~c	e	W	L

(11) **Question:** how do we arrive at **Tableau 4** starting from data that gives less informative tableaux like **Tableau 2** or **Tableau 3**?

(12) **Answer:** (a) ***FUSE*;**
(b) ***RETAIN ALL THE ROWS WHERE INFORMATION IS LOST IN THE FUSION***.

(13) **Fusion** (see Prince 2002a, b):

Fusion	**L**	**e**	**W**
L	L	L	L
e	L	e	W
W	L	W	W

a. **L** is dominant: for any X in {W, e, L}, f(L, X)=L;
b. **e** is identity: for any X in {W, e, L}, f(e, X)=X;
c. f(W, W)=W;
d. fusion of two rows r_1 and r_2 is obtained by **constraintwise** fusion and is abbreviated as: f(r_1, r_2).

Example: Row r_4 in **Tableau 4** is precisely the fusion of **Tableau 2** or **3**.

Tableau 2	**1**	**2**	**3**
r_1: a~b	W	L	W
r_2: a~c	e	W	L

	1	**2**	**3**
f(r_1, r_2)=r_4	**W**	**L**	**L**

Tableau 3	**1**	**2**	**3**
r_3: d~e	W	L	e
r_2: a~c	e	W	L

	1	**2**	**3**
f(r_3, r_2)=r_4	**W**	**L**	**L**

(14) The fusion f(r_1, r_2)=f(r_1, r_3) retains all the information in r_1 or r_3 and 'strengthens' it, i.e. it 'locally' **maximizes information** in rows r_1 and r_3 based on the rest of the tableau.

(15) **Q:** What exactly is **'strengthening' / 'maximizing' information** in a row?
(16) **A:** Obtaining another row that **entails** it.

(17) **<u>Entailment:</u>** A tableau T_1 entails a tableau T_2 iff

{constraint rankings satisfying T_1}$\subseteq${constraint rankings satisfying T_2}

e.g.

Tableau 4	**1**	**2**	**3**
r_4: f~g	W	L	L
r_2: a~c	e	W	L

T_2	**1**	**2**	**3**
	W	L	e
	W	e	L

Tableau 4 entails **T_2** because:
{1>>2>>3} satisfies **Tableau 4**;
{1>>2>>3, 1>>3>>2} satisfies **T_2**.

(18) **However: fusion is not enough!**
The ranking information provided by r_2=eWL, i.e. 2>>3, is lost in the fusion f(r_1, r_2)=WLL, which requires 1>>2 AND 1>>3.

Tableau 2	**1**	**2**	**3**
r_1: a~b	W	L	W
r_2: a~c	e	W	L

	1	**2**	**3**
f(r_1, r_2)=r_4	**W**	**L**	**L**

(19) What we want is to obtain an **equivalent** tableau. **Hence:** we **keep row r_2 together with the fusion f(r_1, r_2)** – which yields precisely **Tableau 4**!

Tableau 2	**1**	**2**	**3**
r_1: a~b	W	L	W
r_2: a~c	e	W	L

	1	**2**	**3**
f(r_1, r_2)	**W**	**L**	**L**

	1	**2**	**3**
r_2	**e**	**W**	**L**

(20) **<u>Equivalence:</u>** A tableau T_1 is equivalent with a tableau T_2 iff
{constraint rankings satisfying T_1}={constraint rankings satisfying T_2}

e.g.

Tableau 2	**1**	**2**	**3**
r_1: a~b	W	L	W
r_2: a~c	e	W	L

Tableau 3	**1**	**2**	**3**
r_3: d~e	W	L	e
r_2: a~c	e	W	L

Tableau 4	**1**	**2**	**3**
r_4: f~g	W	L	L
r_2: a~c	e	W	L

<u>The general problem is:</u>

(21) **Q:** How do we identify the rows that lose information in the fusion?
(22) **A:** *By identifying* ***info loss configurations***.

(23) **Only one case in which no information is lost in the fusion:** when all the W's in the fusion come only from the fusion of W's (i.e. no e's).

Example: Consider **Tableau 11** below. The W in the fusion comes only from the fusion of W's and the fusion **f(r_3, r_6)** is indeed equivalent with the two fused rows.

Tableau 11	**1**	**2**	**3**
r_3	W	L	e
r_6	W	e	L
f(r_3, r_6)	**W**	**L**	**L**

r_3: 1>>2; **r_6:** 1>>3 **f(r_3, r_6):** 1>>2 AND 1>>3

(24) In **Tableau 2**, the fusion lost information because the W in the fusion did not come only from the fusion of W's.

Tableau 2	**1**	**2**	**3**
r_1: a~b	W	L	W
r_2: a~c	e	W	L
f(r_1, r_2)	**W**	**L**	**L**

(25) ***Info loss configuration*** **:=** a constraint **i** in a tableau that contains only **e**'s and at least one **W**.

Example: constraint **1** in **Tableau 2** is the only **info loss configuration** in that tableau.

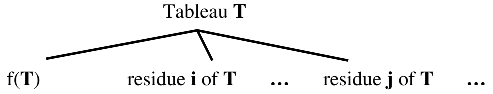
(26) ***Info residue*** **:=** for every info loss configuration **i**, the info residue is the set of rows that have an **e** in that info loss configuration.

Example: the **info residue** corresponding to info loss configuration **1** in **Tableau 2** is r_2=eWL.

(27) Intermediate summary 1 – to obtain the MIB do the following:

- **A. Whole fusion:**
  fuse all tableau rows and construct a branch for the fusion.
- **B. Info Loss:**
  identify all the info loss configurations; for each info loss configuration, construct a branch with the info residue.

(28) The general schema is given in the tree below:



Example:

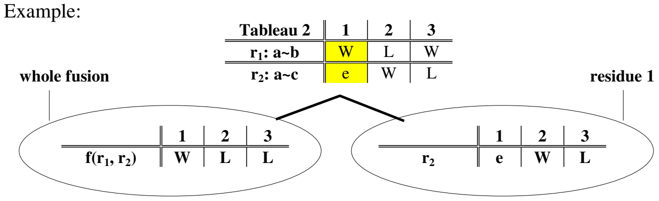


Tableau 2	**1**	**2**	**3**
r_1: a~b	W	L	W
r_2: a~c	e	W	L

	1	**2**	**3**
f(r_1, r_2)	**W**	**L**	**L**

	1	**2**	**3**
r_2	**e**	**W**	**L**

(29) (a) the root of the tree, i.e. **Tableau 2**, is equivalent to its daughters.
(b) the two daughters put together form **Tableau 4**.

(30) A more complex example is provided by **Tableau 12** below: we **recursively** apply the general schema in (28) to all the daughters, until there are no more info loss configurations.

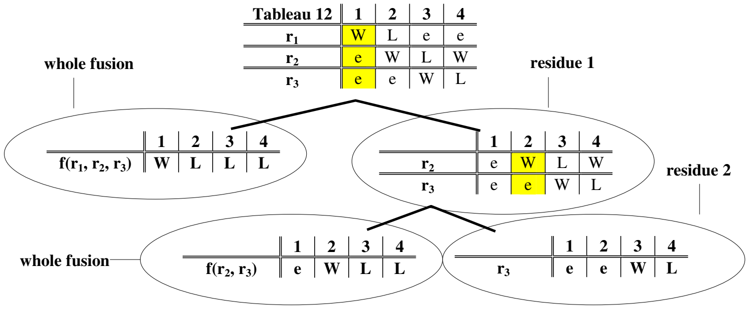


Tableau 12	**1**	**2**	**3**	**4**
r_1	W	L	e	e
r_2	e	W	L	W
r_3	e	e	W	L

	1	**2**	**3**	**4**
f(r_1, r_2, r_3)	**W**	**L**	**L**	**L**

	1	**2**	**3**	**4**
r_2	e	W	L	W
r_3	e	e	W	L

	1	**2**	**3**	**4**
f(r_2, r_3)	**e**	**W**	**L**	**L**

	1	**2**	**3**	**4**
r_3	**e**	**e**	**W**	**L**

(31) Each node of the tree is equivalent to its daughters.

(32) New info loss configurations can appear at the intermediate levels of the tree: constraint **2** is **not** an info loss configuration **at the root level**, but it **is** an info loss configuration at the intermediate level (**residue 1**).

(33) The dynamics of 'disclosing' new info loss configurations is common to **FRed** and **RCD**.

(34) **To obtain the MIB, we just have to collect all the terminal nodes of the tree** – as in **Tableau 13** below.

Tableau 13	**1**	**2**	**3**	**4**
f(r_1, r_2, r_3)	W	L	L	L
f(r_2, r_3)	e	W	L	L
r_3	e	e	W	L

(35) Intermediate summary 2 – to obtain the MIB do the following:

CORE:

CoreA. Whole fusion:
fuse all tableau rows and construct a branch for the fusion.

CoreB. Info Loss:
identify all the info loss configurations; for each info loss configuration, construct a branch with the info residue.

RECURSION:

RecurseA. Recursion:
for every info residue, apply steps (**CoreA-C**).

RecurseB. Terminate:
continue until there are no info residues.

RecurseC. Collect:
collect all the terminal nodes: this is **MIB**(input tableau).

(36) Applying this procedure:
- (a) we obtain a final tableau that is **equivalent** to the initial tableau;
- (b) we **maximize informativeness** in each row of the output tableau;
- (c) we make the resulting tableau **more concise**.

(37) To see that the resulting tableau is as concise as possible, consider again **Tableau 11**. There is **no info loss configuration**, hence we construct only the fusion branch and the resulting tableau contains only one row (as opposed to two).

Tableau 11	**1**	**2**	**3**
r_3	W	L	e
r_6	W	e	L

	1	**2**	**3**
f(r_3, r_6)	**W**	**L**	**L**

(38) **However: fusion is not always useful!** Consider **Tableau 14** below: we construct a branch for each info loss configuration and store the corresponding info residue.

(39) **Fusion is futile in this case: f(r_6, r_2)** is entailed by each of the fused rows **r_6** and **r_2**. Hence: **we should discard the fusion when we collect the terminal nodes**.

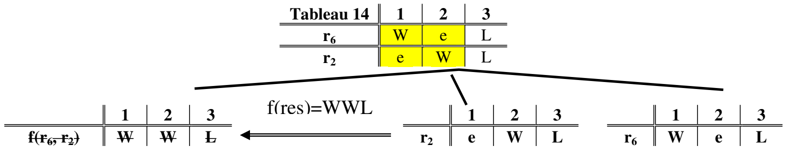


Tableau 14	**1**	**2**	**3**
r_6	W	e	L
r_2	e	W	L

	1	**2**	**3**
~~f(r_6, r_2)~~	~~W~~	~~W~~	~~L~~

f(res)=WWL

	1	**2**	**3**
r_2	**e**	**W**	**L**

	1	**2**	**3**
r_6	**W**	**e**	**L**

(40) The **MIB** of **Tableau 14** is **Tableau 14** itself!

(41) **Hence:** we should add an **Entailment Check** step in the procedure given in (35), so that we eliminate futile fusions, i.e. **we should check whether the residues entail the fusion of the whole**.

(42) **<u>Entailment check:</u>** it (can be proved that it) is sufficient to check whether the fusion of the all the residues entails the fusion of the whole mother node, i.e.

Check: f(All Residue Daughters) entails f(Whole mother)?
If yes, discard f(Whole mother); else, keep it.

(43) THE FUSIONAL REDUCTION (FRED) ALGORITHM:

CORE:

CoreA. Whole fusion:
fuse all tableau rows and construct a branch for the fusion.

CoreB. Info Loss:
identify all the info loss configurations; for each info loss configuration, construct a branch with the info residue.

CoreC. Entailment check:
fuse of all the rows in the info residues constructed in (**CoreB**); if the fusion of residues entails the whole fusion in (**CoreA**), discard the whole fusion in (**CoreA**), else keep it.

RECURSION:

RecurseA. Recursion:
for every info residue, apply steps (**CoreA-C**).

RecurseB. Terminate:
continue until there are no info residues.

RecurseC. Collect:
collect all the terminal nodes: this is **MIB**(input tableau).

2.1. APPLYING FRED: A MORE COMPLEX EXAMPLE.

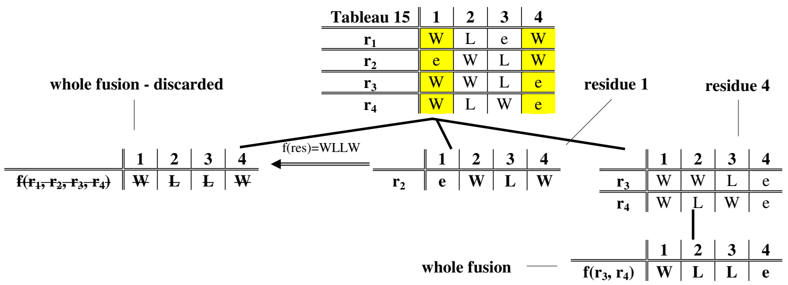


Tableau 15	1	2	3	4
r_1	W	L	e	W
r_2	e	W	L	W
r_3	W	W	L	e
r_4	W	L	W	e

	1	2	3	4
~~$f(r_1, r_2, r_3, r_4)$~~	~~W~~	~~L~~	~~L~~	~~W~~

	1	2	3	4
r_2	e	W	L	W

	1	2	3	4
r_3	W	W	L	e
r_4	W	L	W	e

	1	2	3	4
$f(r_3, r_4)$	W	L	L	e

(44) Applying **FRed** to **Tableau 15**:

CoreA. Whole fusion: we obtain $f(r_1, r_2, r_3, r_4)$=WLLW.

CoreB. Info Loss: there are two info loss configurations: constraint **1** and **4**; for each of them, we construct a branch with the info residue, i.e. residue**1={r_2}** and residue**4={r_3, r_4}**.

CoreC. Entailment check: the fusion of all the info residues is f(res)=WLLW, which is identical to $f(r_1, r_2, r_3, r_4)$ – hence, we discard $f(r_1, r_2, r_3, r_4)$.

RecurseA. Recursion: for every info residue, we apply step (**a**):
residue**1** – $f(r_2)=r_2$ (trivially)
residue**4 –** $f(r_3, r_4)$=WLLe;
there are no more info loss configurations, hence steps (**CoreB**, **CoreC**) do not apply.
RecurseB. Terminate: no more info residues.
RecurseC. Collect: all the terminal nodes are collected in **Tableau 16** below;
this is the **MIB**.

Tableau 16	1	2	3	4
r_2	e	W	L	W
$f(r_3, r_4)$	W	L	L	e

(45) The above example is interesting because:

- we have another example of **reduction / conciseness:** we eliminate the redundant / entailed row r_1 and we collapse rows r_3 and r_4 (that do not create any info loss configuration) into one row that entails both of them.
- we see how a fusion is discarded due to **Entailment Check** (step **CoreC**);
- we see a tree with more than two levels;
- we see an instance of ternary branching, because there are **two** info loss configurations and **two** corresponding residues.

2.2. PROPERTIES OF THE FRED.

- **FRed eliminates entailed rows.**
- **FRed collapses all the rows that do not create any info loss configuration (i.e. W-compliant rows – see Prince 2002a)**
- **FRed incorporates maximal information into each derived row.**
- **FRed and Recusive constraint demotion (RCD):** the algorithm can perform RCD – instead of constructing a branch for each info residue (**CoreB**), intersect all the info residues and construct **only one branch for this intersection**; the fusions that are collected at the end are the fusions based on which RCD extracts the constraint ranking.
- **FRed detects inconsistency:** if it is applied to an inconsistent (contradictory) tableau (see (46) below), then the output of the algorithm is going to contain a row with at least one L and no W's (hence, an unsatisfiable row).
- **FRed detects the maximal set of inconsistent rows:** for an inconsistent tableau, the output of FRed contains a row with at least one L and no W's; this row appears at some node in the tree; the mother of that node is the corresponding maximal set of inconsistent rows.

(46) **A contradictory / inconsistent tableau** is not satisfied by any constraint ranking.

Contradictory	1	2	3
r_1	L	W	e
r_2	W	L	e
$f(r_1, r_2)$	L	L	e

3. CONCLUSION.

The present paper posed **the following question**: given that a comparative tableau is a (partial) characterization of the constraint ranking of a particular language, how can we know:

(4) what are the necessary and sufficient ranking conditions enforced by a tableau, i.e. what are the constraint rankings that satisfy it?

The proposed solution: for each (non-contradictory) comparative tableau T, there is a unique canonical tableau called **Maximally Informative Basis (MIB)** such that:

(a) MIB(T) is **equivalent** to T;
(b) MIB(T) contains a **minimal number** of **independent** rows;
(c) each row in **MIB**(T) is **maximally informative**.

The **MIB** answers question **(4)**: it displays in a **perspicuous / maximally informative** way all the necessary and sufficient ranking conditions.

Finally, **the Fusional Reduction (FRed) algorithm** outputs the **MIB** for any (non-contradictory and non-trivial) input tableau. In addition, **FRed**:

(a) can identify if the input tableau is contradictory and the minimal set of rows in the input tableau that yields a contradiction;
(b) can perform Recursive Constraint Demotion (**RCD**).

ACKNOWLEDGEMENTS.

The authors, whose names are arranged alphabetically, would like to thank Jane Grimshaw, Naz Merchant, Paul Smolensky, and Bruce Tesar for useful discussion; and, for helpful comments, audiences at the Rutgers Optimality Research Group (April 2004, Nov. 2005), HUMDRUM (May 2004), and the LSA 2005 Summer Institute. For support during aspects of the investigation, we thank the John Simon Guggenheim Memorial Foundation, the National Science Foundation (Grant No. BCS-0083101) and the National Institutes of Health (NRSA Training Grant No. 1-T32-MH-19975-05). *The opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or of the National Institutes of Health.* The authors also claim responsibility for any errors, omissions, misprisions or infelicities that may have crept into the text.

REFERENCES.

Brasoveanu, Adrian & Alan Prince 2005. *Ranking and Necessity*, ROA 794, http://roa.rutgers.edu/view.php3?id=1094.

Brasoveanu, Adrian 2002. The Interaction of Word Stress and Phrasal Nuclear Pitch Accent in Chickasaw, http://www.rci.rutgers.edu/~abrsvn/pdfs/OT/phon_term_paper.pdf.

Brasoveanu, Adrian 2003. *Minimal Fusion Normal Form*, ms. RU/NB, http://www.rci.rutgers.edu/~abrsvn/pdfs/OT/MFNF.pdf.

Grimshaw, Jane 1997. Projection, Heads, and Optimality, *Linguistic Inquiry* 28.4, 373-422; ROA-68.

Merchant, Nazarré 2004 FRed: a Java implemenation., ms. RU/NB.

Prince, Alan 2002a. *Entailed Ranking Arguments*, **ROA-500**.

Prince, Alan 2002b. *Arguing Optimality*, to appear in Coetzee, Andries, Angela Carpenter and Paul de Lacy (eds). *Papers in Optimality Theory II*. GLSA, UMass. Amherst; **ROA-562**.

Tesar, Bruce & Paul Smolensky 1993. The Learnability of Optimality Theory: an Algorithm and some Basic Complexity Results. ROA-2.

Tesar, Bruce & Paul Smolensky 2000. *Learnability in Optimality Theory*. MIT Press.

Tesar, Bruce 1995. *Computational Optimality Theory*. Ph. D. Dissertation, University of Colorado at Boulder. ROA-90.

CHICKASAW ERC SET (ADRIAN BRASOVEANU, 2002) – 14 CONSTRAINTS, 41 ROWS:

X	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**	**C13**	**C14**
1	W	W	e	L	W	W	L	e	e	W	W	e	e	e
2	e	e	e	L	W	e	L	W	W	W	W	e	e	e
3	e	W	e	e	e	W	e	L	e	e	e	e	e	e
4	e	W	e	e	W	e	L	L	e	W	W	e	e	e
5	e	e	e	e	W	e	L	e	W	W	W	e	e	e
6	e	e	e	L	W	e	L	W	W	W	W	W	e	e
7	e	L	e	e	e	L	e	W	W	e	e	e	e	e
8	e	e	W	e	e	e	W	e	e	L	L	e	e	e
9	e	L	e	W	e	L	W	W	e	L	L	W	e	e
10	e	L	e	e	W	L	e	W	W	e	e	W	e	e
11	e	L	e	e	W	L	W	W	W	L	L	W	e	e
12	e	e	e	e	e	W	L	e	e	W	e	e	e	e
13	e	L	e	e	e	e	e	e	W	e	e	e	e	e
14	e	L	e	e	e	e	L	e	W	W	e	e	e	e
15	e	L	e	W	L	e	e	e	e	e	L	e	e	e
16	e	L	e	W	L	e	L	e	e	e	L	e	e	e
17	e	L	e	W	L	e	L	e	W	W	e	e	e	e
18	e	e	W	e	e	e	e	e	e	e	L	e	e	e
19	W	W	e	L	W	e	e	e	e	e	W	e	e	e
20	e	e	e	W	L	e	W	L	e	e	e	e	e	e
21	e	e	e	W	e	e	e	L	e	e	e	e	e	e
22	e	e	e	W	W	e	L	L	W	W	W	e	e	e
23	e	e	e	e	W	e	L	e	W	W	W	e	e	e
24	e	e	e	L	W	e	L	W	W	W	W	e	e	e
25	e	e	e	L	W	e	e	W	W	e	e	e	e	e
26	W	e	e	e	e	W	e	L	L	e	e	L	e	e
27	e	e	e	e	e	e	W	e	e	L	e	e	e	e
28	e	e	e	e	e	e	W	e	e	L	e	e	e	e
29	e	e	e	e	e	L	W	W	e	L	e	W	e	e
30	e	e	e	e	e	L	W	W	e	e	e	e	e	e
31	e	e	e	e	e	L	e	W	e	e	e	W	e	L
32	e	e	e	e	e	W	L	e	e	W	e	e	e	e
33	e	e	e	e	e	W	L	e	e	e	e	e	e	e
34	e	e	e	e	e	e	W	e	e	L	e	e	e	e
35	e	e	W	L	e	e	e	e	e	e	e	e	e	e
36	e	e	e	L	W	e	L	W	W	W	W	W	e	e
37	e	e	e	L	W	e	e	W	W	e	W	e	e	e
38	e	e	e	e	e	L	e	W	e	e	e	e	e	L
39	e	W	e	L	e	e	e	e	e	e	e	e	W	L
40	e	e	e	e	e	e	W	e	e	L	L	e	e	L
41	e	e	W	L	e	e	W	e	e	L	L	e	e	L

REDUCED TO MIB VIA THE IMPLEMENTATION OF FRED IN MERCHANT 2004 – 9 ROWS:

X	C1	C13	C3	C9	C4	C2	C8	C6	C7	C12	C5	C14	C10	C11
f1	**W**	*e*	*e*	L	L	L	L	L	L	L	L	L	L	L
f2	*e*	**W**	*e*	*e*	L	L	L	L	L	*e*	L	L	L	L
f3	*e*	*e*	**W**	*e*	L	L	L	L	L	*e*	L	L	L	L
f4	*e*	*e*	*e*	**W**	L	L	L	L	L	*e*	L	L	L	L
f5	*e*	*e*	*e*	*e*	**W**	L	L	L	L	*e*	L	L	L	L
f6	*e*	*e*	*e*	*e*	*e*	**W**	L	L	L	*e*	*e*	L	L	L
f7	*e*	*e*	*e*	*e*	*e*	*e*	**W**	L	L	*e*	*e*	L	L	L
f8	*e*	*e*	*e*	*e*	*e*	*e*	*e*	**W**	L	*e*	*e*	L	L	L
f9	*e*	*e*	*e*	*e*	*e*	*e*	*e*	*e*	**W**	*e*	*e*	L	L	L

Input data:	41 rows
MIB:	9 rows

Avg. disjunctivity of data:	2.7 W's/row	Max disjunctivity of data:	6 W's in one row
Avg. disjunctivity **MIB**:	1 W/row	Max disjunctivity **MIB**:	1 W/row
Avg. conjunctivity of data:	1.8 L/row	Max conjunctivity of data:	4 L's in one row
Avg. conjunctivity **MIB**:	7.1 L/row	Max conjunctivity **MIB**:	11 L's in one tow

Summary:

- nearly **5:1** tableau shrinkage;
- nearly **3:1** reduction in avg. disjunctivity;
- nearly **1:4** increase in avg. conjunctivity.