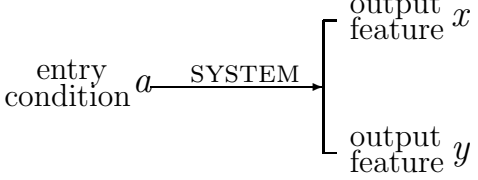
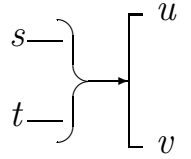
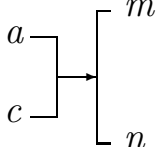
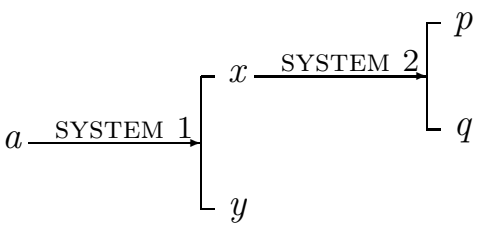
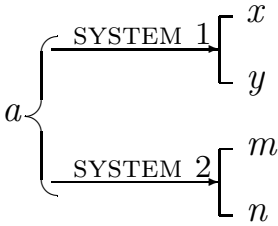
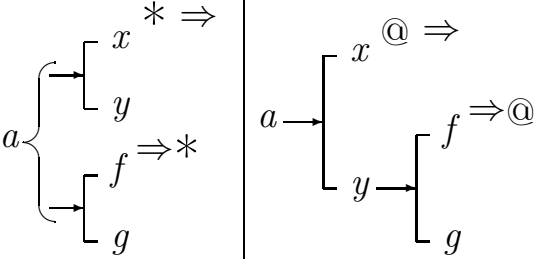
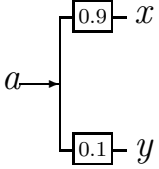


## A.10 System network conventions (1)

 <p>entry condition <math>a</math> — SYSTEM — <math>\left\{ \begin{array}{l} \text{output feature } x \\ \text{output feature } y \end{array} \right.</math></p>	<p><b>System:</b>          ‘if <math>a</math>, then either <math>x</math> or <math>y</math>’.          Possible feature selection expressions:  <math>ax, ay</math>.</p>
 <p><math>\left. \begin{array}{l} s \\ t \end{array} \right\}</math> — <math>\left\{ \begin{array}{l} u \\ v \end{array} \right.</math></p>	<p><b>Conjunct entry condition:</b>          ‘if both <math>s</math> and <math>t</math>, then either <math>x</math> or <math>y</math>’.          Possible feature selection expressions:  <math>s, t, stu, stv</math>.</p>
 <p><math>\left. \begin{array}{l} a \\ c \end{array} \right\}</math> — <math>\left\{ \begin{array}{l} m \\ n \end{array} \right.</math></p>	<p><b>Disjunct entry condition:</b>          ‘if either <math>a</math> or <math>c</math>, then either <math>m</math> or <math>n</math>’.          Possible feature selection expressions:  <math>am, an, cm, cn</math>.</p>
 <p><math>a</math> — SYSTEM 1 — <math>\left\{ \begin{array}{l} x \\ y \end{array} \right.</math>  <math>x</math> — SYSTEM 2 — <math>\left\{ \begin{array}{l} p \\ q \end{array} \right.</math></p>	<p><b>Systems ordered in delicacy:</b>          ‘if <math>a</math>, then either <math>x</math> or <math>y</math>,          and if <math>x</math>, then either <math>p</math> or <math>q</math>’.          Possible feature selection expressions:  <math>axp, axq, ay</math>.</p>
 <p><math>a</math> — <math>\left\{ \begin{array}{l} \text{SYSTEM 1} \\ \text{SYSTEM 2} \end{array} \right.</math> — <math>\left\{ \begin{array}{l} x \\ y \\ m \\ n \end{array} \right.</math></p>	<p><b>Simultaneous systems:</b>          ‘if <math>a</math>, then both either <math>x</math> or <math>y</math>          and, independently, either <math>m</math> or <math>n</math>’.          Possible feature selection expressions:  <math>axm, axn, aym, ayn</math>.</p>
 <p><math>a</math> — <math>\left\{ \begin{array}{l} x^* \\ y \\ f \Rightarrow^* \\ g \end{array} \right.</math>      <math>a</math> — <math>\left\{ \begin{array}{l} x^@ \\ y \\ f \Rightarrow^@ \\ g \end{array} \right.</math></p>	<p><b>Conditional marking:</b>          In either case (<math>*</math> or <math>@</math>):          ‘if <math>x</math>, then <math>f</math>’.          Possible feature selection expressions:  <math>axf, ayf, ayg</math>.</p>
 <p><math>a</math> — <math>\left\{ \begin{array}{l} 0.9 - x \\ 0.1 - y \end{array} \right.</math></p>	<p><b>Inherent probability:</b>          ‘probability of choosing <math>x</math> is 90%’.          Predicted feature selections (1000 traversals): <math>ax</math> (900 times), <math>ay</math> (100 times).</p>

## A.11 System network conventions (2)

	<p><b>Recursive system</b></p> <p>‘if <math>a</math>, then  both  either <math>z</math> or <math>y</math> or <math>x</math>  and, independently,  either ‘stop’ (<math>  </math>) or ‘go on’ (i.e., re-enter)’.</p> <p>Possible feature selection expressions:</p> <p><math>ax  </math>, <math>ay  </math>, <math>az  </math>,  <math>axx  </math>, <math>axy  </math>, <math>axz  </math>,  <math>ayx  </math>, <math>ayy  </math>, <math>ayz  </math>,  <math>azx  </math>, <math>azy  </math>, <math>azz  </math>,  <math>axxx  </math>, <math>axxy  </math>, <math>axxz  </math>,  <math>axyx  </math>, <math>axyy  </math>, <math>axyz  </math>,  <math>axzx  </math>, <math>axzy  </math>, <math>axzz  </math>,  <math>ayxx  </math>, <math>ayxy  </math>, <math>ayxz  </math>,  ...</p>
	<p><b>Realization statement:</b></p> <p>‘if <math>x</math>, perform realization(s) stipulated’.</p> <p>Here: insert Function1 into structure  (of unit being generated).</p>

## A.12 Realization operators

(Table based on a classification by Christian Matthiessen.)

Type		Name of Operator	Operator Symbol in Use	
Structure-building (presence of function, constituency, ordering)	Presence	Insert	+ Function	
	Constituency	Expand	Function1 ( Function2 )	
		Ordering	Partition	Function1   Function2
			Order	Function1 ^ Function2
			OrderAtFront	§ ^ Function
			OrderAtEnd	Function ^ #
Layering	Conflate	Function1 / Function2		
Inter-rank realization (via preselection in lower-rank networks)	Grammatical preselection	Preselect	Function:feature	
		Subcategorize	Function1::Function2	
	Lexical pre-selection	Lexicalize	Function= <u>lexical item</u>	
		Classify	Function∈class	
		Outclassify	Function∉class	
	Partial preselection	IncreaseProbability	↑p{:feature}{0,5→0,9}	
		DecreaseProbability	↓p{:feature}{0,5→0,1}	

## A.13 Functional grammatical analyses: examples

Helmut Kohl	went	to	Halle
			Fokus
←			Neues
Thema	Rhema		
Subjekt	<i>'did'</i> Finites	<i>'go'</i> Prädikator	Adverbialbestimmung
Modusteil		Restteil	
Beteiligter:: Medium/ Täter:: Reisender	Prozeß::  Handlung:: Fortbewegung	Begleitumstand:: Ort:: Endpunkt:: Reiseziel	

where he was greeted by the citizens

	Fokus
Gegebenes → ←	Neues

Thema	Rhema			
WH-/ Adv.best.	Subjekt	Finites	Prädikator	Adverbialbestimmung
Rest-	Modusteil		-teil	
Bgl.umstand:: Ort	Bteilgtr:: Medium/ Ziel	Prozeß:: Handlung	Beteiligter:: Agens/ Täter	

They threw eggs at the Chancellor

	Fokus	
Gegebenes → ←	Neues	Gegebenes

Thema	Rhema			
Subjekt	<i>'did'</i> Finites	<i>'throw'</i> Prädikator	Komplement	Adverbialbestimmung
Modusteil		Restteil		
Beteiligter:: Agens/ Täter	Prozeß:: Handlung:: Ziel	Beteiligtes:: Medium/ Ziel	Begleitumstand:: Ort:: Endpunkt	

The police arrested the rascals.

	Fokus
←	Neues
Gegebenes	

Thema	Rhema		
Subjekt	<i>'did'</i> Finites	<i>'arrest'</i> Prädikator	Komplement
Modusteil		Restteil	
Beteiligter:: Agens/ Täter	Prozeß:: Handlung:: Ziel	Beteiligtes:: Medium/ Ziel	