Fundamentals of CJK Encoding

LU Qin
Rapporteur of
ISO/IEC JTC1/SC2/WG2/IRG
Dept. of Computing, HK Polytechnic Univ.
csluqin@comp.polyu.edu.hk

Outline

- Introduction to Encoding Methods
  - Principles and Procedures
- Unified CJK Encoding
  - Rules and procedures
  - Examples
- Character description and decomposition
- Latest IRG development

Basic Concepts

- Character Set: A a collection of indivisible symbols.
  - For example, {a, b, c, …z, A, B, C, …, Z, 0, 1, 2, …, 9} is an English character set, or {啊, 阿, 唉, …, 作, 坐, 座} is a Chinese character set.
  - A named set: Ex:大漢和辞典(Daikanwa Dictionary)
  - Each character is unique (Mathematical set)

    ⇨ Members in a set has no ordering in mathematical sense
- Closed set vs. open set
  - English alphabets is a closed set, whereas Chinese characters is an open set by nature.
    - A character set for computer processing is a named set with a finite number of characters *n*.

Coded Character Set

- Coded Character Set (Codeset)
  - a character set in which every character is given a unique computer code so that all characters in the set can be processed by computer systems.
- Encoding
  - Refers to the process of assigning each character a code. Sometimes, we also call this as enumerating the symbols of the character set
    - How each character is distinguished from another
    - How to distinguish different codesets
      - ASCII vs. JIS

Formal definition

- Given a character set, *C*, a coded character set, *CC*, is a set of 2-tuples, $CC=\{(c_i, code_i) \mid c_i \in C, code_i \in CODE\}$, where $code_i \neq code_j$ if $c_i \neq c_j$.
  - Example: *C*={中, 文, 计, 算},
  - $CODE_1$={00, 01, 10, 11},
  - $CODE_2$={0000, 0001, 0010, 0011}

CC_1= {(中, 00), (文, 01), (计, 10), (算, 11)}

CC_2 = {(中, 11), (文, 10), (计, 01), (算, 00)}

CC_3= {(中, 0000), (文, 0001), (计, 0010), (算, 0011)}

$CC_1 \neq CC_2 \neq CC_3$ Why?

Consider a code sequence: 0011

- Code Space selection:
  - Consider the number of characters needs to be supported
    - For English, one byte (i.e. 8 bits), which can provide 256 (i.e. 2^8) code points, is sufficient.
    - For Chinese, since there are more than 256 characters, at least 2 bytes (i.e. 2^{16}=65,536 code points) are necessary
  - A codeset may not use all the code points in a code space, i.e. some are assigned to characters, others are unassigned
- Code Space may take values from different data ranges and the code points are not necessarily all of the same fixed length, Example: {00 – 7F, 8000 – FEFE}

ASCII Code (8×16 Table)

low-bits

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

high-bits

*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

Extended ASCII

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`	128	80	Ç	160	A0	á	192	C0	└	224	E0	α
1	01	Start of heading	33	21	!	65	41	A	97	61	a	129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
2	02	Start of text	34	22	"	66	42	B	98	62	b	130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
3	03	End of text	35	23	#	67	43	C	99	63	c	131	83	â	163	A3	ú	195	C3	├	227	E3	π
4	04	End of transmit	36	24	$	68	44	D	100	64	d	132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
5	05	Enquiry	37	25	%	69	45	E	101	65	e	133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
6	06	Acknowledge	38	26	&	70	46	F	102	66	f	134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
7	07	Audible bell	39	27	'	71	47	G	103	67	g	135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
8	08	Backspace	40	28	(	72	48	H	104	68	h	136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i	137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j	138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k	139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l	140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m	141	8D	ì	173	AD	¡	205	CD	═	237	ED	ø
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n	142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o	143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
16	10	Data link escape	48	30	0	80	50	P	112	70	p	144	90	É	176	B0	░	208	D0	╨	240	F0	≡
17	11	Device control 1	49	31	1	81	51	Q	113	71	q	145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
18	12	Device control 2	50	32	2	82	52	R	114	72	r	146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
19	13	Device control 3	51	33	3	83	53	S	115	73	s	147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
20	14	Device control 4	52	34	4	84	54	T	116	74	t	148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u	149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v	150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
23	17	End trans. block	55	37	7	87	57	W	119	77	w	151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
24	18	Cancel	56	38	8	88	58	X	120	78	x	152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
25	19	End of medium	57	39	9	89	59	Y	121	79	y	153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z	154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{	155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|	156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}	157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~	158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	⌂	159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	□

ISO-8859 series: http://en.wikipedia.org/wiki/ISO_8859

- GB for Simplified Chinese: up to 94x94 (8,836) chars.
  - High byte: 0xA1-0xFE, low byte: 0xA1 - 0xFE
  - Total of 6,773 Chinese characters and 682 other symbols
- Big5 for Traditional Chinese: up to 94x157 (14,758) characters
  - High byte: 0xA1-0xFE, low byte: 0x40-0x7E and 0xA1 - 0xFE
  - Total of 13,052 Chinese characters and 441 other symbols
- JIS standard for Japan

Character vs. glyph

- Character: A unit of a written language that can be used as a non-divisible symbol: A vs B
- Glyphs: represent the shapes that characters can have when they are rendered or displayed.
- Example: A, *A*, are the same character and having the same code. Concrete shape can be very different and are given one codepoint.
- Coding of variants under same or different code points?

Problems with Different Chinese Codesets (locale dependent codesets)

- Codeset incompatibility: difficult to do conversion
  - 1-N mapping, example: 后(gb) vs 后後(big5)
  - 1-0 mapping: some characters in B5 is not in GB
- Different writing styles(simplified and traditional) cannot be presented in the same system
  - switching mechanisms is needed when multiple codesets need to co-exist on the same platform
- Problem with data exchange: Wrong interpretation of data from non-conforming platforms.
- Different software must be developed for different codesets

ISO 10646: UCS-4

(Canonical form of ISO 10646)

- Fixed 31-bit coding assignment(High-bit off)
- 00 00 00 00 to 7F FF FF FF

Group No. (total: 128)	Plane No (total: 256)	High Byte (total: 256)	Low Byte (total: 256)

- Each plane: 2^{16} = 65,536 code points
- **BMP(the basic multilingual plane), ISO 10646-1**
  - Both Group No. and Plan No. are 00(first two bytes of zeros)
- Before ISO 10646 part 2 came out(end of year 2001), only BMP contains characters

Universal Code Set

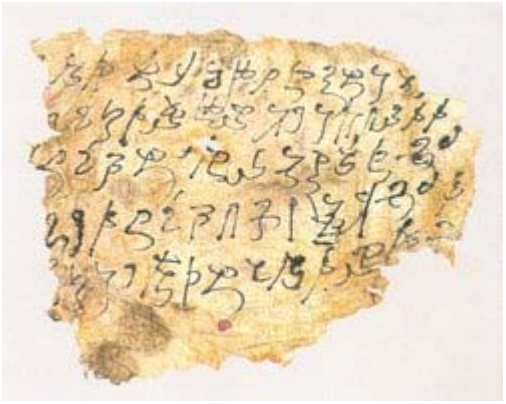
- ISO/IEC 10646 UCS2 and Unicode:up to 256x256 (65,536) characters
  - High byte: 0x00-0xFF, low byte: 0x00 - 0xFF
  - Different characters are put into different zones
  - 20,902 ideograph characters + 6,582 characters in Extension A
- Design Principle: One coding standard for all
- Features:
  - Universal: characters in almost all national standards
  - Framework: Fix the coding architectures, and code-points can be filled up later.
  - Uniform and Efficient: fixed-width encoding, no need to identify the coding length(ASCII, Big5, GB)
  - Unambiguous: Any given 16-bit(32-bit) value always represents the same character

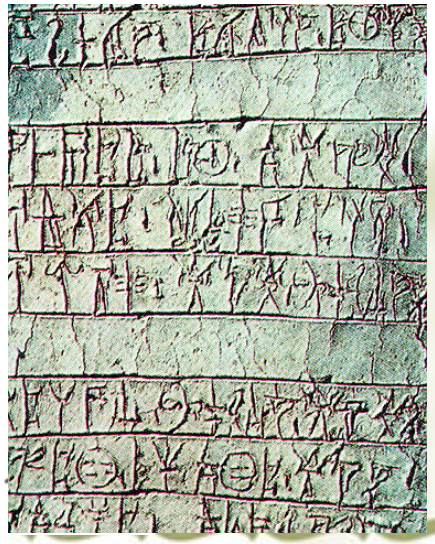
ISO 10646-2

Plane 1, the Supplementary Multilingual Plane, (SMP) is mostly used for historic scripts such as Linear B, but is also used for musical and mathematical symbols.

Plane 2, the Supplementary Ideographic Plane (SIP), is used for about 40,000 rare Chinese characters that are mostly historic, although there are some modern ones.

Plane 14, the Supplementary Special-purpose Plane (SSP), currently contains some non-recommended language tag characters and some variation selection characters.





Need for CJK Unification

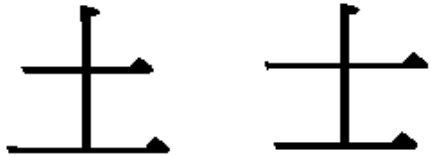
- Unification Problems:
  - Different sources
    - What would be considered the same character even if the glyphs are different
- Three-dimensional Conceptual Model:
  semantics(x), abstract shape(y), actual shape(z)
- Examples:
  田儿贤一 vs.田见贤一 vs. 田兒賢一vs.田見賢一?
  TAGO Kenichi

- Semantics requires reference to dictionaries:
  康熙字典(Kanxi),大漢和辭典(Daikanwa), etc..

Unification Rules(認同規則)

- R1: Source Separation Rule: If two ideographs are distinct in a primary source standard, then they are not unified.
  - Less useful in future ext.

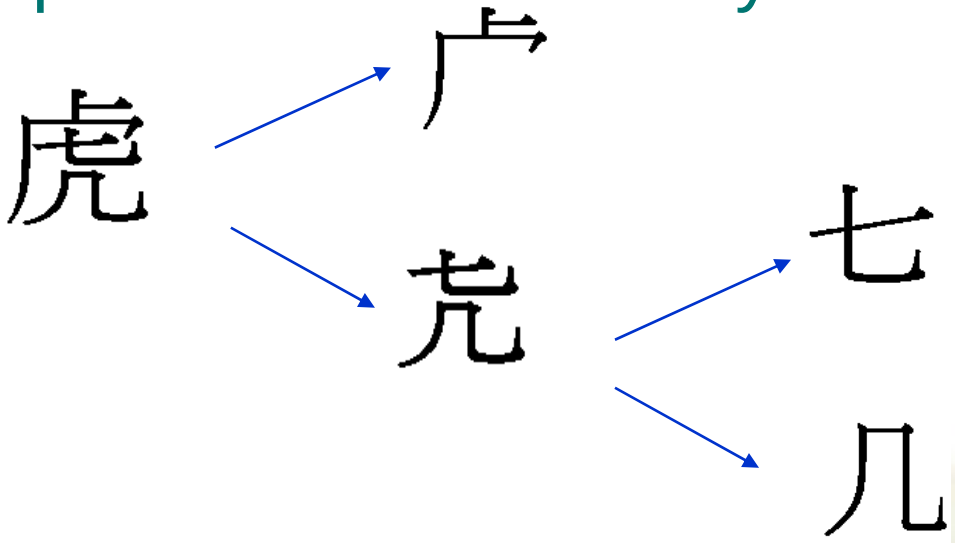
- R2: Non-cognate(非同源)Rule: In general, if two ideographs are unrelated in historical derivation(non-cognate characters), then they are not unified

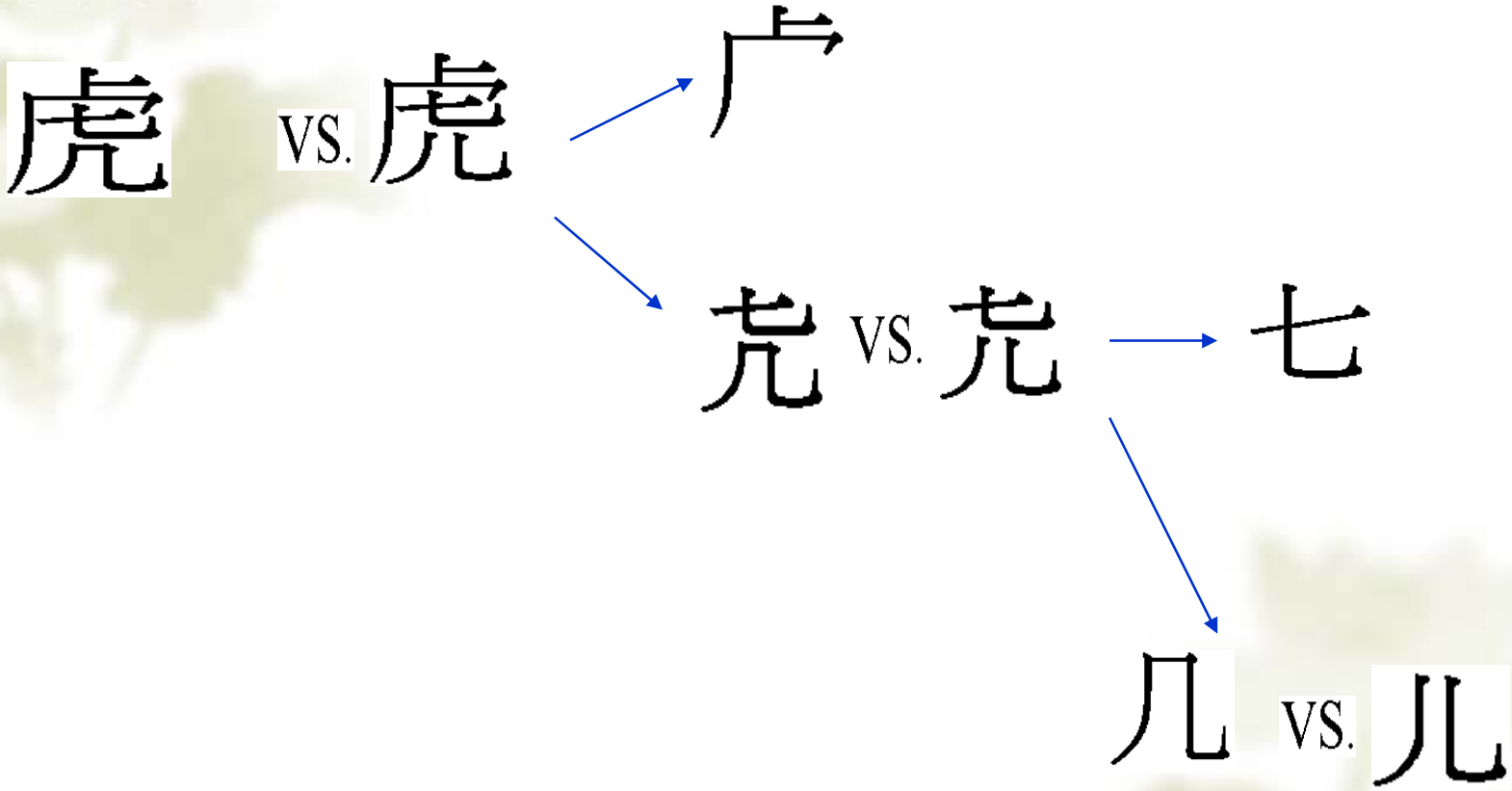


- R3: By means of two-level classification, the abstract shape of each ideograph is determined. Any two ideographs that possess the same abstract shape are unified unless disallowed by R1 or R2.

Character decomposition

- Example 虎 虎
- Component structure analysis





- **Abstract shape looks at decomposition structure first**

Code chart example of UCS2

ISO/IEC 10646:2003 (E)

Row/Cell Hex Code	C G-Hanzi-T	J Kanji	K Hanja	V ChuNom
053/176 **35B0**			3-225B 3-0259	
053/177 **35B1**			3-225C 3-0260	
053/178 **35B2**			3-225D 3-0261	
053/179 **35B3**			3-225E 3-0262	
053/180 **35B4**	3-3853 3-2451			
053/181 **35B5**	3-3650 3-2248	4-306F 4-1679		
053/182 **35B6**	5-3676 5-2286	A-216E A-0178		
053/183				

Row/Cell Hex Code	C G-Hanzi-T	J Kanji	K Hanja	V ChuNom
053/192 **35C0**	3-366A 3-2274	4-3660 4-2264	3-2260 3-0264	
053/193 **35C1**	5-3879 5-2489	3-3A3D 3-2629	A-216F A-0179	3-2261 3-0265
053/194 **35C2**		4-365F 4-2263		0-324E 0-1846
053/195 **35C3**	5-392A 5-2510	3-3A37 3-2623		
053/196 **35C4**	3-3668 3-2272	6-4739 6-3925		
053/197 **35C5**	S-2356 S-0354	5-3679 5-2289		
053/198 **35C6**		5-367A 5-2290	A-2170 A-0180	
053/199				

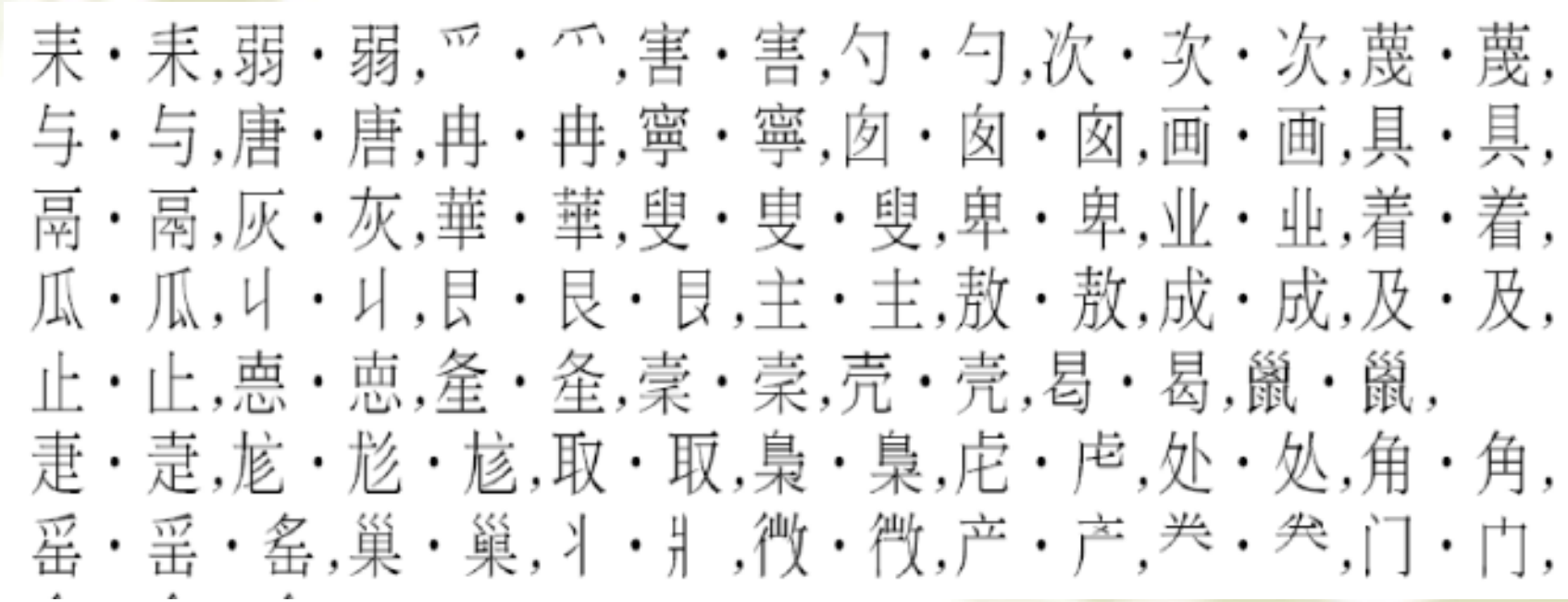
Row/Cell Hex Code	C G-Hanzi-T	J Kanji	K Hanja	V ChuNom
053/208 **35D0**	3-3A3A 3-2626			
053/209 **35D1**	3-366B 3-2275			0-3238 0-1824
053/210 **35D2**	3-3671 3-2281			
053/211 **35D3**	3-366E 3-2278	5-3E39 5-3025		
053/212 **35D4**	3-367B 3-2291	4-3C4A 4-2842	A-2172 A-0182	
053/213 **35D5**	3-3678 3-2288	4-3C46 4-2838		
053/214 **35D6**	3-404F 3-3247	A-2173 A-0183		0-3231 0-1817
053/215				

Examples of Annex S use

- Implication of Source Separation Rule:
  - If not for source separation, they should be considered unified
  - As components, they are unifiable

丟丢	T	兌兑	T	翍翀	T	吶吶	T
4E1F 4E22		514C 5151		524F 5259		5436 5450	
么幺	GT	兎兔	TJ	剝剥	T	告告	T
4E48 5E7A		514E 5154		525D 5265		543F 544A	
爭爭	GTJ	兗兖	T	劒劒	J	唧唧	T
4E89 722D		5156 5157		5292 5294		5527 559E	
仞仞	J	冊冊	TJ	勻匀	T	喻喻	T
4EDE 4EED		518A 518C		52FB 5300		55A9 55BB	

- Additional unification examples



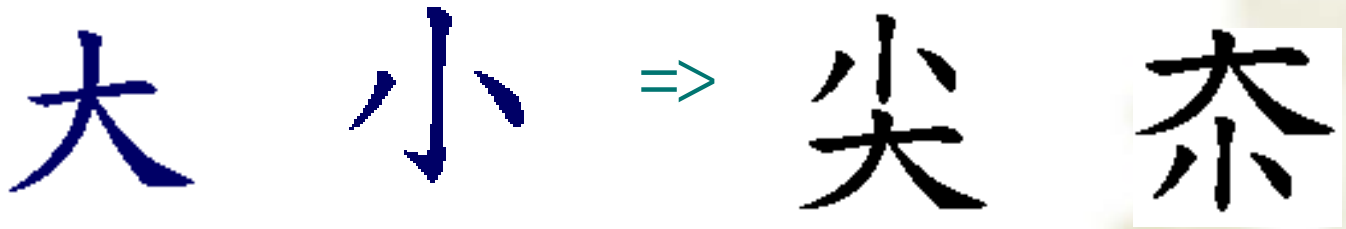
- Still continuing work

Problems with ideograph Character Encoding

- Each character is treated as a different symbol, and thus given a code point: variations?
- Code point assignment in a block does try to follow radical order, but codepoint assignment does not consider the substructures(components). Thus such information is not revealed.
- When a new character is created, code point allocation is needed, potentially endless standardization process
- Encoding of rarely used ideograph characters is a waste of resource both in terms of code space and also standardization effort

Characteristics of Ideographs

- Ideograph characters are often formed by smaller ideographic elements such as Radicals, ideographs proper, and other ideographic components
- Natural in the formation of characters
- Examples: 2 components



Chinese has long been using components to describe characters, especially characters with the same pronunciation

Character Structure Analysis

- Use of ideograph description characters
  - 12 IDCs to describe character structures

											
2FF0	2FF1	2FF2	2FF3	2FF4	2FF5	2FF6	2FF7	2FF8	2FF9	2FFA	2FFB
left-to-right	above-to-below	left-middle-right	above-middle-below	overall-around	Down-to-Encom-pass	up-to-emcom pass	right-to-encom-pass	right-down-encompass	left-down-encom-pass	right-up-encom-pass	Embed-ment

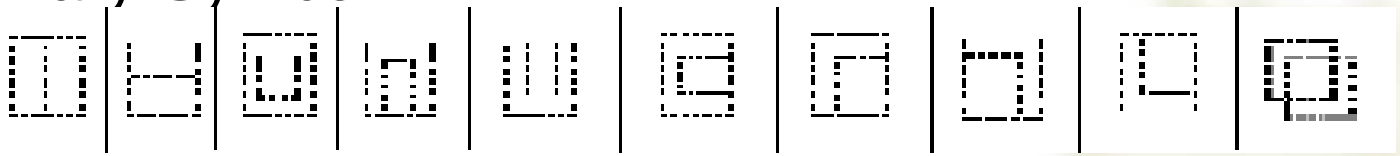
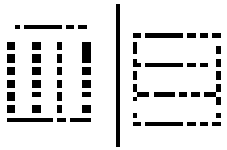
- Ideograph description sequence
  - Method of using both IDCs and component characters to describe a character

IDS

- IDS describes a character using its components and indicating the relative positions of the components.
- IDCs are considered operators to the components.
- IDSs can be expressed by a context free grammar through the Backus Naur Form. The grammar G has four components:
- Let G = {Σ, N, P, S}, where
  - Σ: the set of terminal symbols-coded radicals, coded ideographs, and the 12 IDCs.
  - N:the set of 5 non-terminal symbols

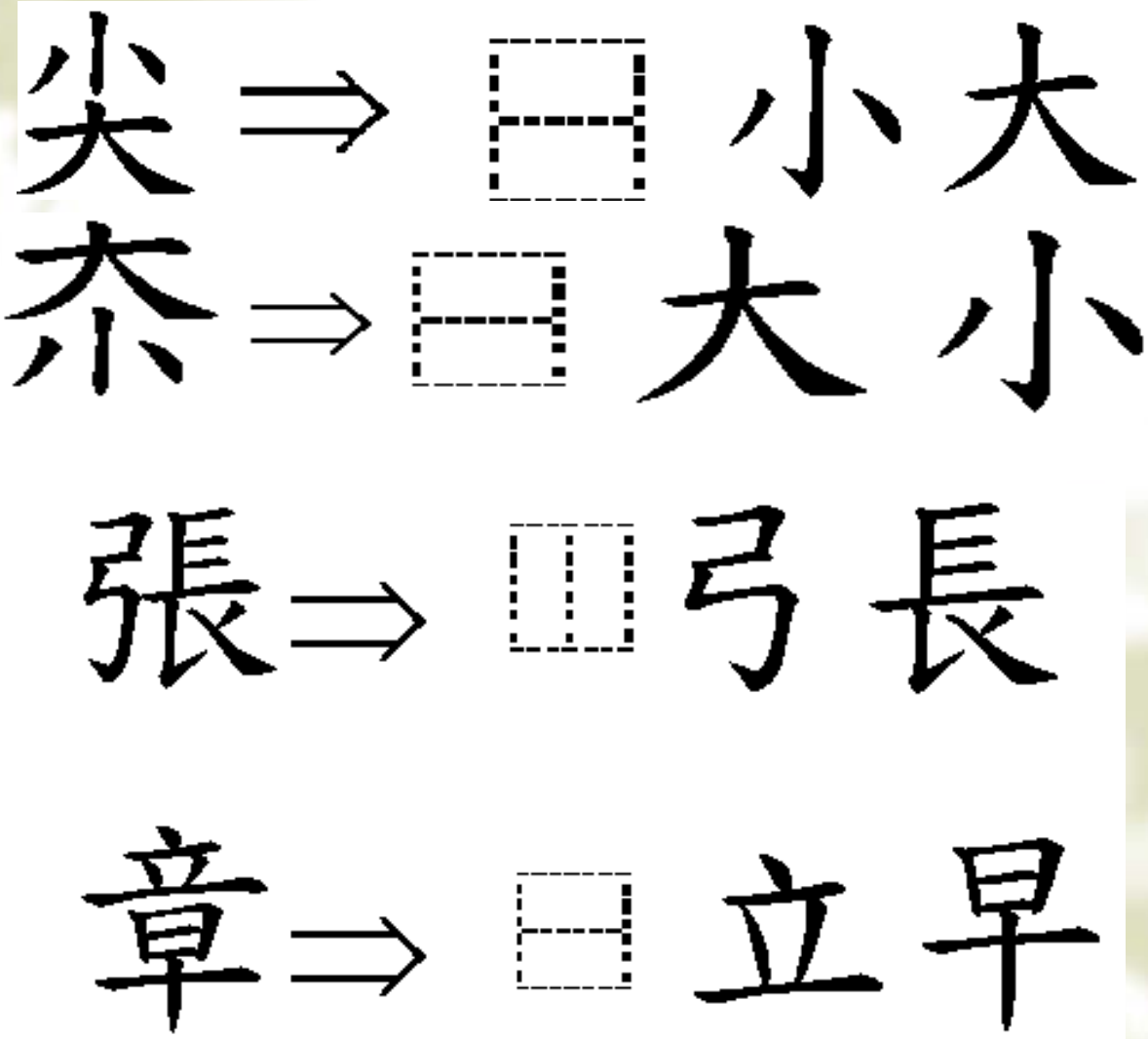
    N={IDS, IDS1, Binary_Symbol, Ternary_Symbol, Ideograph_Component}
  - S = {IDS}, which is the start symbol of the grammar
  - P: a set of rewrite rules

- IDS::=<Binary_Symbol><IDS1><IDS1>|<Ternary_Symbol>

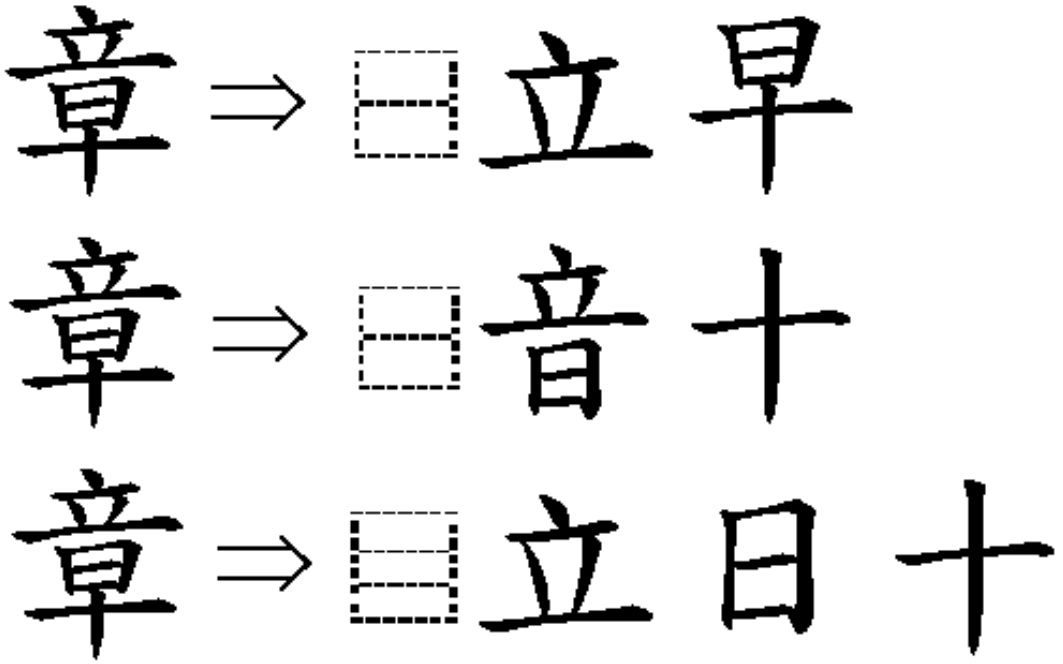
  <IDS1><IDS1><IDS1>
- <IDS1> ::= <IDS> | <Ideograph_Component>
- <Ideograph_Component>::= coded_ideograph | coded_radical | coded_component
- <Binary-Symbol> ::= 
- <Ternary_Symbol> ::= 

- Note that even though the IDCs are terminal symbols, they are not part of the ideograph components.

Examples

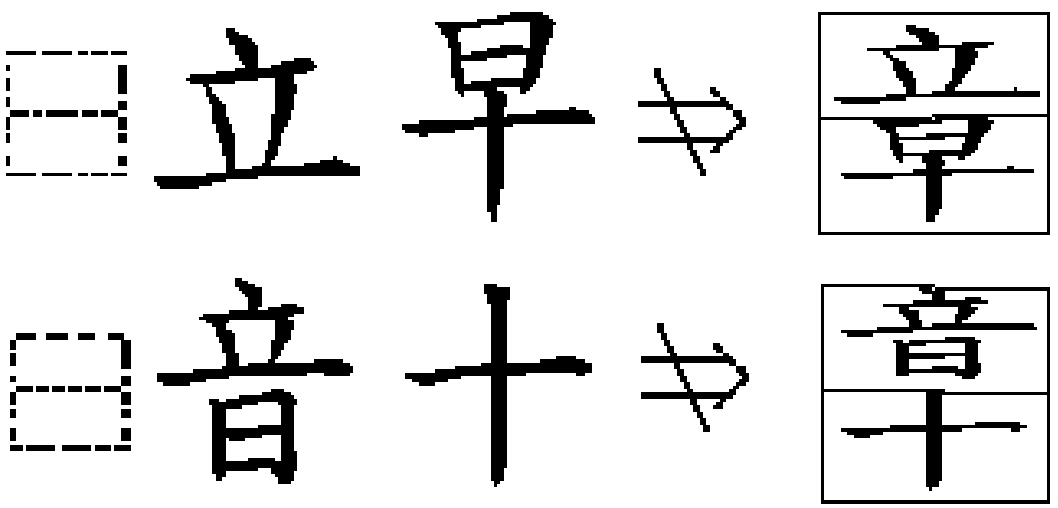


- IDS allows a character to be described by different sequences



- Additional rules and checking still needed by IRG(IRGN 1183 on IDS)

- IDS describes ideographic character composition at the abstract level. It indicates the relative positions of the components, but does not indicate the proportions.
- Not intended for rendering.
- Nesting is natural in ideographs and they are reflected in the IDS scheme



Extending the Objectives of IDCs

- Using coded characters to describe not yet code ideographs both for representation and exchange

- Limit standardization to only modern characters, and not some rarely used characters
- Learning of character composition(education)
- Revealing substructures of ideograph characters
- Description of ideograph variants

Components

- Ideographic Components(IRG definition):
  units which can be used to represent ideographs. These components consist of ideographs proper coded in ISO 10646 (BMP) and some basic elements used to form ideographs.
- Radicals(IRG definition): those ideographic components listed in index pages of KX(China), DKW(Japan), DJW(Korea), HYD(China)
- ISO extensions:
  - Radicals
  - Components

More examples in IRGN 1183 on IDS

Internet Client/Server Model

- Overview of HTTP connection:
  - Open connection
  - Request for service
  - Response from server
  - close connection
- An 8-bit clean protocol, ensuring safe transmission of all forms of data including Chinese
- New features from HTTP/1.1 --- data type negotiation
- Codeset announcement in request message

(1) Codeset announcement in request message

Accept-charset, Accept-language

(2) Codeset announcement in response message

Content-type, Content-language

HTML

- Earlier version of HTML has no mechanism to tell data are written in what codeset, everything defaults to ISO-8859-1.
- New features of HTML from Version 3.0:

  A new tag <LANG> is designed to tell what codeset the document is written in , for example:

  <META HTTP-EQUIV="Content-Type" CONTENT="text/html";CHARSET=big5">

  <LANG=gb2312>

  ... …/* tagging at each segment */

  </LANG>

  Tag <LANG> makes automatic codeset identification of web documents possible.
- Default CHARSET=ISO10646-1:1993 Not ISO8859-1:1998

Conclusion

- Computer coding moves towards international standards, ISO10646
  - Can include all character sets
  - Avoided locale dependent codesets
    - Universal: easier for processing, exchange
  - Technical issues to solve
  - Too many characters can create problems