

基于 BNF 的 3DS 文件格式规范化描述^①

陈 蕾, 赵正旭

(石家庄铁道大学 信息科学与技术学院, 石家庄 050043)

摘 要: 3DStudio(3DS)是 3dsMax 用来保存模型信息的一种文件结构, 是业界的通用标准格式之一. 为了使不同星系之间的模型可以被自由调用, 解决可视化系统中模型格式不兼容的问题, 减小 3DS 文件的存储空间, 提出使用巴科斯范式(BNF)规范化描述 3DS 文件. 描述了 BNF 的特点和对 3DS 文件结构进行表述和分析的使用方法, 并以土卫三模型的 3DS 文件为例, 使用 BNF 对其进行形式化定义, 从而形成了统一的规范化管理格式, 以利于搜索引擎的检索和应用.

关键词: 3DS; BNF; 形式化语言; 规范化

3DS File Format Specification Based on BNF

CHEN Lei, ZHAO Zheng-Xu

(School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang 050043, China)

Abstract: 3DStudio (3DS) is a file structure of 3dsMax used to store model information, which is one of the industry's standard format in 3D software. In order to ensure the models of different system can be called freely, to solve the problem that the different model format is not compatible in visual system, and to reduce the storage space of 3DS file, this paper proposes to use the Backus-Naur Form (BNF) specification to describe the 3DS file. This paper describes the characteristics of the BNF and the method of describing and analyzing the 3DS file structure, and takes the 3DS file of Tethys model as an example, using BNF to formally definite it, thus forms a unified format of standardization management, which facilitates retrieval and application of search engine.

Key words: 3DS; BNF; Formal language; specification

随着虚拟现实技术的发展, 基于三维模型的深空探测实时三维可视化技术需求越来越大, 对三维模型的需求也随之增大. 目前, 常用的三维建模软件包括 AutoCAD^[1]、3DS Max^[2]、Autodesk Maya^[3]、Zbrush^[4]等, 不同的三维建模软件适用于不同的应用领域, 生成的模型格式也各种各样. 在深空探测实时三维可视化系统中, 需要存储成千上万的三维模型, 由于模型的数据格式和文件结构不同, 可能会导致各个星系互不兼容, 使得这些三维模型不能在同一实时系统中同时被调用, 导致在深空探测实时可视化系统中不能快速精确地调用目标模型. 如果对系统中三维模型的文件格式和描述方式制定统一的规范, 对三维模型文件

进行标准化和规范化处理, 就会大大提高系统的效率.

文件标准化和规范化是实现三维模型资源共享与重用的前提条件, 在资源建设中占有极其重要的地位. 规范化管理三维模型的途径之一就是模型的文件结构“标准化”. 首先统一三维模型的文件格式, 本文选择 3DS^[5]格式的三维模型, 它是一种比较通用的保存三维模型信息的文件格式. 其次需要对 3DS 文件进行规范化描述, 制定一个统一的描述规则. 目前对于 3DS 的文件格式和块结构都是以树状图或自然语言描述, 必须用自然语言加以说明才能将 3DS 的文件结构描述清楚. 这些图表和自然语言计算机识别不了, 现有的文本描述方式和自然语言解释方式存在二义性等

^① 基金项目:河北省高层次人才科学研究项目(GCC2014010)

收稿时间:2016-08-10;收到修改稿时间:2016-09-13 [doi:10.15888/j.cnki.csa.005721]

弱点, 因此必须寻找一种规范化的形式将其描述.

描述文件数据结构的常用语言有 XML (eXtensible Markup Language)^[6]、JSON(JavaScript Object Notation)^[7]、Z 语言^[8,10]、BNF^[11]. XML 具有文档结构清晰、容易阅读等优点, 但其文件尺寸一般都比较大, 因此在现有的网络条件下传输文件效率较低^[12]. JSON 采用完全独立于语言的文本格式, 易于人阅读和编写, 同时也易于机器解析和生成. Z 语言是一种用“数学文字”或“数学符号”来描述计算机系统的规范化语言, 是软件工程中编码之前的规格说明语言. BNF 是一种上下文无关文法的符号集, 由于其简洁、明了和科学而被广泛接受^[13,14].

经过比较, BNF 具有简单易学、开放、易于扩展、分析效率高等优点, 相对于其它形式化语法, 在描述

文件格式方面具有更好的适用性. 因此, 本文使用 BNF 描述 3DS 文件结构, 统一描述文件格式, 以一种计算机可识别的文法将其规范化描述, 将 3DS 文件变为“标准”格式, 建立一种专有的数据格式, 使其更加通用.

1 3DS文件解析

3DS 文件是树状结构的二进制文件, 基本单位是块, 每个块都有各自的用途. 所有块都是以树形结构组织的, 以一个主块开始, 以嵌套的形式构成.

文件的主块 ID 是 0x4D4D, 这可以判断一个文件是否是 3DS 文件. 3DS 文件的每个块都是一个层次结构, 每一部分都有各自特有的 ID 号, 图 1 表示了 3DS 文件中块的存在结构及位置(文件太长, 只画出文件前几个块, 其他不再赘述).

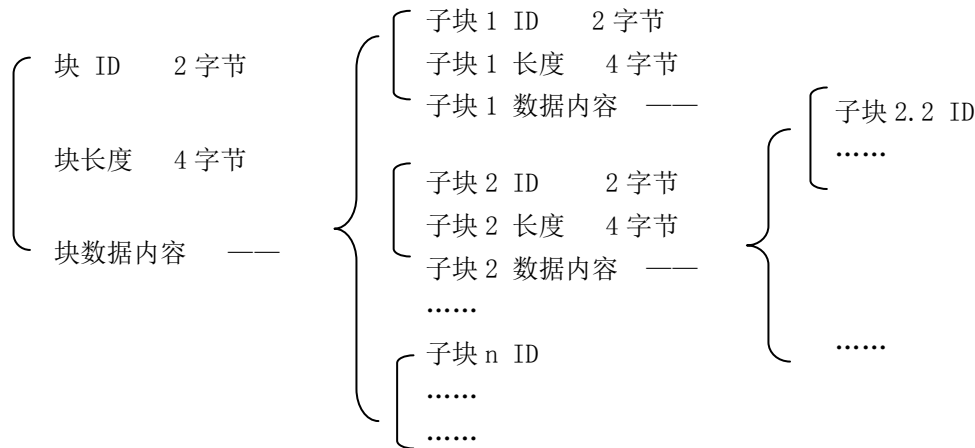


图 1 3DS 文件结构和块

这些“块”描述了 3DS 文件的数据信息, 由块头和数据两部分组成. 块头包括 ID 和长度两项, 共占 6 位, 长度指得是整个块的长度, 而不只是数据的长度. 每

一个“块”都是以 2 个字节的 ID 开始, 紧接着是 4 个字节的块长度信息, 紧接着是块数据信息^[15,16], 如图 2 所示.

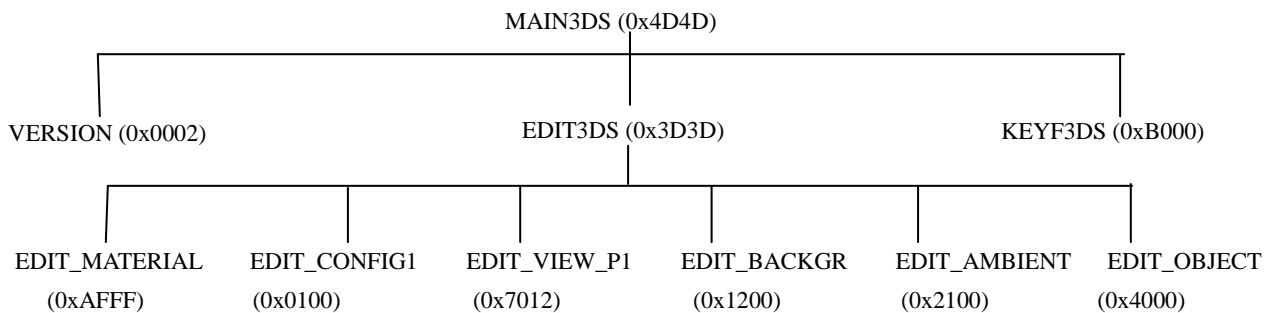


图 2 3DS 文件块的嵌套

从图 1 和图 2 可以看出, 3DS 文件是一层套一层的树状结构, 而且这个树状结构是一个无比庞大的家族, 每一个块都包含着自己的子块, 子块接着包含子块, 如此下去.

2 3DS文件的BNF文法表示

2.1 BNF 范式

BNF 就是使用一种类似的形式来描述一个问题, 让所有参与者都能明白他们对哪一方面能够达成一致意见. 也就是对于某个问题制定一个统一的标准, 为某个符号赋予一个特定的意义, 在某个领域内它就只能是这个意思, 别人在了解这个标准和所制定的规则之后, 就能理解使用这种符号集的表达方式和内容.

BNF 被广泛应用于编程语句, 也可以用来规范自然语言^[17,18]. BNF 非常精确, 在语法上没有歧义; 形式简单易学, 便于使用者和计算机处理; 开放、易于扩展、表达能力较强, 能充分表达出描述者意图. 而用普通语言描述的语法不仅冗长, 而且会造成理解上的歧义; 用 BNF 表示的语法是数学产物, 可以被计算机理解, 因此选择 BNF 描述文件语法结构.

2.2 块结构描述

文章上一节中, 使用自然语言和图表的形式对 3DS 的文件结构进行了描述, 这种方式虽然通俗易懂, 但是由于结构比较繁杂, 没有形成一个统一的形式. 如果使用 BNF 制定一种通用的文法, 结果会更加一目了然, 计算机也更容易识别. 对图 1 和图 2 中的 3DS 文件结构制定 BNF 文法, 则表示为:

(1) 首先对一些专业术语作以下定义:

块 ::= chunk

子块 ::= sub-chunk

块头 ::= head

数据 ::= data

长度 ::= size

(2) 句子文法如下:

3DS file ::= <chunk>{chunk}

chunk ::= <head><data>

head ::= <ID><size>

data ::= <data>{sub-chunk}

sub-chunk ::= chunk

由于 3DS 文件的块 ID 是由十六进制数表示的, 2 个字节, 一共四位; size 表示的是整个块的长度, 也是

用十六进制数表示的, 所以其 BNF 表示文法如下:

ID ::= "0x"<Digit | Capital> <Digit | Capital>
<Digit | Capital> <Digit | Capital>

size ::= "0x"<Digit | Capital> <Digit | Capital>
<Digit | Capital> <Digit | Capital><Digit | Capital><Digit | Capital><Digit | Capital><Digit | Capital><Digit | Capital><Digit | Capital><Digit | Capital>

Digit ::= <0|1|2|3|4|5|6|7|8|9>

Capital ::= <A|B|C|D|E|F>

BNF 范式中有不定循环的规则^[19], 格式为: *Rule, 即在元素前加操作符 * 表示重复, 完整形式为: <a>*element, 例如: 3*3<element>表示该元素只允许 3 次, 所以块 ID 和 size 可以简化为:

ID ::= "0x", 4*4<Digit | Capital>(ID 是两个字节, 4 位表示, 即重复 4 次)

size ::= "0x", 8*8<Digit | Capital> (size 是四个字节, 8 位表示, 即重复 8 次)

每一个块的 ID 都不一样, 都有自己的专属表示, 比如主块 ID 是 0x4D4D, 编辑块是 0x3D3D, 具体块的 ID 只需要根据实际情况灵活替换即可.

通过上述几个文法表示, 采用递归法可以简洁地表示出 3DS 文件中块与块的包含关系和树状结构. 这种使用字母和符号表达的文法没有二义性, 计算机更容易识别, 更通用.

2.3 文件描述

在描述 3DS 文件的时候, 如果单纯从实体外观上描述, 可能会因为一些语言共识或者描述方式的不同, 造成理解上的歧义. 另一方面, 每个 3d 模型都有自己的特点, 都描述出来就会出现很多语句, 这些语句可能只是换了一个形容词, 或者只是换了一个名词而已, 这样就会做许多重复的工作. 如果, 现在用一种规范定义出 3DS 文件的描述形式, 那就会省很多工作量, 只需要套用形式就行.

通过上述 3DS 文件的描述, 我们可以了解到 3DS 文件数据格式是由“块”组成的, 它是一个树形层级结构, 主块包含子块的关系.

首先, 一个 3DS 文件的开始是主块 MAIN3DS, 其 ID 是 0x4D4D, 它是基本信息块, 所以在一个 3DS 文件中, 主块是必选项, 它总是以主块 0x4D4D 开始的. 其次是一级子块, 版本信息块 VERSION, ID 是 0x0002, 编辑信息块 EDIT3DS, ID 是 0x3D3D, 关键帧

信息块 KEY3DS, ID 是 0xB000.

最基本的语句规范描述如下:

```
3DS File ::= <MAIN3DS> + <VERSION> +
<EDIT3DS> + [KEYF3DS]
```

因为一个 3DS 文件总是以主块开始的, 所以 <MAIN3DS>是必选项, 用尖括号括住以表示必选. 紧接着是版本信息块、编辑信息块和关键帧块. 关键帧信息块是关于帧动画信息的数据, 所以这一部分内容不是必须存在的, 所以用[]括起来表述可选项内容.

编辑块的父块是 0x4D4D, 其主要子块是 EDIT_MATERIAL 材质块, ID 是 0xAFFF; EDIT_OBJECT 对象块, ID 是 0x4000.

```
EDIT3DS::=[EDIT_MATERIAL]+<EDIT_CONFI
G>+<EDIT_VIEW_P3>+<EDIT_BACKGR>+<EDIT_O
BJECT>{+EDIT_UNKNWN}
```

```
EDIT_MATERIAL::=<EDIT_NAME01>+<MAT_A
MBCOL>+<MAT_DIFCOL>+<MAT_SPECOL>+<MATM
AP>
```

```
EDIT_VIEW_P3::=<TOP>+<BOTTOM>+<LEFT>
+<RIGHT>+<FRONT>+<BACK>+<USER>+<CAMER
A>+<LIGHT>+[DISABLED]+[BOGUS]
```

```
EDIT_OBJECT::=<OBJ_TRIMESH>+<OBJ_LIGH
T>+<OBJ_CAMERA>+{+OBJ_UNKNWN}
```

```
OBJ_TRIMESH::=<TRI_VERTEXL>+<TRI_VER
TEXOPTIONS>+<TRI_MAPPINGCOORA>+<TRI_M
APPINGSTANDARD>+<TRI_FACEL1>
```

```
OBJ_LIGHT::=<LIT_OFF>+<LIT_SPOT>+<LIT_
UNKNWN01>
```

```
OBJ_CAMERA::=<CAM_UNKNWN>+{+CAM_U
NKNWNN}
```

```
KEYF3DS::=<KEYE_UNKNWN01>+<EDIT_VIE
W1>+<KEYF+FRAMES>+<KEYF_OBJDES>+{+KEYF
_UNKNW}
```

```
KEYF_OBJDES ::=<KEYF_OBJHIERARCH>+<K
EYF_OBJDUMMYNAME>+<KEYF_OBJTRANSLAT
E>+<KEYF_OBJROTATE>+<KEYF_OBJSCALE>+{+K
EYF_OBJUNKNWN}
```

EDIT_UNKNWN、OBJ_UNKNWN 等指的是一些未知信息块, 它可以有多个, 也可以没有, 所以用花括号括起来, 表示重复 0 到无数次.

主编辑块的子块虽然是按照一定的次序存放, 但

其中有些块并不是一定存在的, 比如: 如果你没有定义材质, 使用缺省材质, 这里将不存在材质块.

2.4 文件数据结构描述

2.4.1 宏定义

在一个 3DS 文件中, 使用宏定义来定义文件中的块:

```
#define MAIN3DS 0x4D4D //定义主块
```

使用 BNF 统一描述为:

```
Macro definition::= “#define”, <Name><ID> //具
体块的 Name 根据实际情况替换即可
```

2.4.2 定义数据结构

在 3DS 文件中, 保存块信息的结构如下:

```
struct CLscChunk
```

```
{
```

```
unsigned short int ID; //块的 ID
```

```
unsigned int length; //块的长度
```

```
unsigned int bytesRead; //需要读的块数据的字节数
```

```
}
```

使用 BNF 描述为:

```
struct_statement::=“struct CLscChunk”“{”<variable_
declaration>“}”
```

3 实例分析

3.1 土卫三.3DS 文件数据格式解析

许多文献已经介绍过, 3DS 文件的二进制位存储顺序是 Intel 式的^[20], 低字节在前, 高字节在后, 所以在读取 3DS 数据文件的时候必须考虑这个规则. 下面通过一个具体实例对 3DS 文件的数据格式进行详细解析: 首先, 通过 3DSMax2010 创建土卫三的三维模型, 以.3DS 的格式导出并保存, 如图 3 所示.

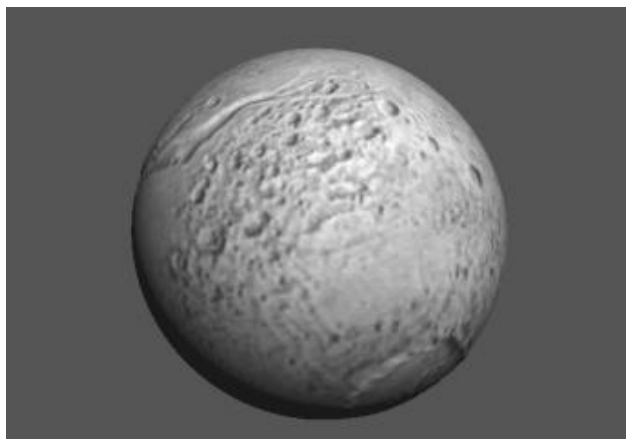


图 3 3dsMax 打开土卫三.3DS 的三维模型

通过工具 C32asm 读取图 3 中 .3DS 的土卫三文件, 得到其十六进制数据列表, 如图 4 所示. 图中所显示的数据以两个字节的块 ID 开始, 之后是四个字节的块的长度信息. 从数据中提取到标志性的块 ID, 比如 0x4D4D(主块)、0x3D3D(编辑块)及 0xAFFF(材质块)等, 它们是两个字节的块 ID, 后面紧跟 4 个字节该块的长度信息.

图 4 3DS 格式的土卫三十六进制数据

从土卫三 3DS 文件中截取一组数据如下:

4D 4D CA C9 09 00 02 00 0A 00 00 00 03 00 00 00
3D 3D DE C8 09 00.....

(1) 首先, 4D4D 是该文件的主块 ID, 占两个字节.

(2) 之后的 4 个字节保存的是主块的长度信息, 也就是整个文件的长度. 由于 3DS 二进制数据是按低位在前、高位在后存储的, 所以主块的长度为 0x0009C9CA.

(3) 接着是该 3DS 文件的第一个子块的 ID, 0x0002. 子块的长度是 0x0000000A.

(4) 之后的块 ID 未知, 归为未知信息, 可以跳过

(5) 0x3D3D 是主编辑块的 ID. 该块的长度为 0x0009C8DE.

(6) 之后的字节的读取规则按如上步骤.

3.2 土卫三 3DS 文件的 BNF 描述

根据以上的研究内容, 运用 BNF 描述土卫三.3DS 文件结构, 具体内容如下:

```
3DS File::=<chunk>{+chunk} ①
chunk::=<ID><size><data | sub-chunk> ②
sub-chunk::=chunk ③
```

根据式①和图 4 中土卫三 3DS 文件的十六进制序列, 其 BNF 描述规范如下:

土卫三.3DS ::= <MAIN3DS>

根据式②和③扩展如下:

(1) 基本块的表示:

MAIN3DS::="0x4D4D" "0009C9CA"

+ (<VERSION> + <EDIT3DS>)

(2) 版本块的表示:

VERSION::="0x2000" "0000000A" "00000003"

(3) 编辑块的表示:

EDIT3DS::="0x3D3D" "0009C8DE"+(<EDIT_MATERIAL>+<EDIT_OBJECT>{+EDIT_UNKNWN})

(4) 材质块的表示:

EDIT_MATERIAL::="0xAFFF" "000000E4"+(<MAT_NAME01>+<MAT_AMBCOL>+<MAT_DIFCOL>+<MAT_SPECOL>+<MAT_SHININESS>+<MATMAP >)

MAT_NAME01::="0xA000" "00000013" "746C7561666544202D2D3130"

MAT_AMBCOL::="0xA010" "00000F" "969696000090011"

MAT_DIFCOL::="0xA020" "00000F" "969696000090011"

MAT_SPECOL::="0xA030" "00000F" "E5E5E5000090011"

MAT_SHININESS::="0xA040" "000000E" "000A00000080030"

MATMAP::="0xA200" "00000032" "0064000000080030"

(5) 综合以上所有式子, 总式表示为:

土卫三.3DS ::= "0x4D4D" "0009C9CA"+ "0x2000" "000000A" "0000003"+ "0x3D3D" "0009C8DE"+ ("0xAFFF" "000000E4"+ ("0xA000" "00000013" "746C7561666544202D2D3130"+ "0xA010" "00000F" "969696000090011"+ "0xA020" "00000F" "969696000090011"+ "0xA030" "00000F" "E5E5E5000090011"+ "0xA040" "000000E" "000A00000080030"+ "0xA200" "00000032" "006400000080030")+<EDIT_OBJECT>{+EDIT_UNKNWN})

由于 3DS 文件是嵌套结构, 因此以上是通过递归的方式表示 3DS 的 BNF 描述方法, 先用非终结符一层

一层表示,直到结构最中心再用终结符一层一层替换.由于土卫三.3DS文件块结构特别庞大,其中有许多未知块和不重要的块,文章不再赘述,只选取了其中比较重要的块.上述所用数据是从土卫三.3DS文件的十六进制数据序列中摘取,数据属实.

从上述式子可以看出,这种描述方法与图4中文文件的十六进制数据相比,更清晰、更简洁.省去了没有用的信息之后,所有表述的都是模型文件必要的特征信息,减小了文件存储空间.

4 结论与展望

如何将知识提炼成规则,需要用到统一的文法.文法最吸引人的特性,是它们自然地描述了所定义语言句子的分层语法结构.基于BNF的3DS文件文法表示,可以更科学地描述文件的结构,更加简洁、通用.

在深空探测三维可视化系统中,将模型文件格式统一,保证在各个星系之间模型可以被自由调用,解决了不同星系系统中模型格式不兼容的问题.使用巴克斯范式描述3DS文件,可以缩小文件长度,同时在描述过程中,把文件中不重要的部分省略,描述文件的必要信息,在不影响模型正常显示的情况下,将文件描述到最简,这不仅有助于计算机快速读取,也方便使用者识别.

文章为更好地管理和使用3DS文件提出一种新思路,希望能对改进3DS文件的存储空间和传送速率起到一定的借鉴作用.接下来,需要对深空探测系统中大量天体模型的具体描述,在应用中完善该描述方法,能够进一步促进航天事业的发展.另外,计划在3DS文件分类过程中使用BNF描述代码,制定统一的描述规范,如何利用BNF对3DS文件制定分类规范也是下一步要进行的研究工作.

参考文献

- 1 百度百科.AutoCAD.http://baike.baidu.com/view/25199.htm.[2016-07-27].
- 2 百度百科.3dsmax.http://baike.baidu.com/view/11137.htm#reference-[1]-11137-wrap.[2016-08-01].
- 3 百度百科.Autodesk Maya.http://baike.baidu.com/view/6463073.htm?fromtitle=maya&fromid=38497&type=syn.[2016-07-15].
- 4 百度百科.ZBrush.http://baike.baidu.com/view/165599.htm.[2016-08-05].
- 5 杜琳,蒋辉.VC++中利用DirectX实现3DS文件的读取和控制.计算机时代,2010,01:41-43.
- 6 搜狗百科.XML.http://baike.sogou.com/v34402.htm?fromTitle=XML.[2016-07-26].
- 7 百度百科.JSON.http://baike.baidu.com/view/136475.htm.[2016-07-27].
- 8 赵正旭,温晋杰,赵卫华.Z规范及其使用方法.北京:科学出版社,2015.
- 9 Wen JJ, Guo Y, Zhao ZX. Representation of raspberry PI practice in Z notation. British Journal of Applied Science & Technology, 2016, (4): 2231-0843.
- 10 赵正旭,温晋杰.Z规格说明自动生成器.计算机系统应用,2016,25(4):148-155.
- 11 百度百科.BNF.http://baike.baidu.com/view/1137652.htm.[2016-07-24].
- 12 程小东,张宇.X3D文件转换为JSON的方法.池州学院学报,2009,23(3):29-30.
- 13 Zhao W, Xia CH, Luo Y, Liu XC, Wu WK. An approach for description of computer network defense scheme and its simulation verification. Journal of Computers, 2014, (2): 388-398.
- 14 罗晨,鲍广宇,刘晓明,袁黎苗.基于改进BNF的作战管理语言语法形式化方法.计算机科学,2012,39(4):189-192.
- 15 刘芳,刘贤梅.3DS文件读取、绘制与控制方法的研究与应用.计算机工程与设计,2009,30(19):4575-4578.
- 16 百度文库.Autodesk公司官方3ds文件格式介绍.http://download.csdn.net/detail/whucv/4851225.[2015-11-15].
- 17 Xiong W, Jin YH, Liu ZY. Recognizing Chinese number and quantifier prefix to enhance statistical parser in machine translation. Journal of Computers, 2014, (4): 867-874.
- 18 Laros JFJ, Blavier A, den Dunnen JT, Taschner PEM. A formalized description of the standard human variant nomenclature in Extended Backus-Naur Form. BMC Bioinformatics, 2011, (12): S5.
- 19 绿色背影.巴科斯范式. http://xl-feng1314.blog.163.com/blog/static/4450715220080196551349/. [2016-05-15].
- 20 刘爽,张恒博.三维建模软件3dsMax数据文件3ds的解析.大连民族学院学报,2012,14(3):260-264.