인공지능기초응용 II

3 주차 과제

인공지능응용

K2025029 금동환

목차

1. 아래 용어에 대한 설명 및 차이점을 설명하시오.

   - KS완성형 코드
     - 초성, 중성, 종성을 나누지 않고 완성된 글자 단위로 표현
     - 한글 11,172(초성(19),중성(21),종성(28) 의 조)자 중에서 사용빈도가 낮은 것들은 제외하고 2,350자 정의
     - 8,822(11,172-2,350)자는 완성형코드에 정의되지 않음 (문제 있음)
   - KS X 1001(KS C 5601)
     - 8,822자 (한글 2,350 + 한자 4,888 + 기타 1,584)
       - KS완성형 코드 한글 2,350자
       - 교육용 한자, 이름에 사용되는 한자 등 4,888자
       - 기타 (원문자, 괄호문자, 문장부호 등) 1,584 자
   - cp949
     - 11,172자 (2,350 + 8,822)
     - 완성형에 누락된 8,822자 사용 가능.
       - KS완성형 코드로는 2,350자에 포함되지 않은 글자는

         똠->또ㅁ, 펩시콜라->페ㅍ시콜라 으로 표현되므로 문제 있음.
       - ms에서 자사제품을 문제없이 사용하도록 KS완성형 코드에 포함되지 않은 한글을 포함시킨 코드셋.
       - ms949 라고도 함.
   - EUC-KR
     - Unix 환경에서 한글을 표현하기 위한 방식 (Extended Unix Code-KR)
     - KS완성형에 누락된 8,822자 표현하지 못함.

2. 유니코드와 관련하여

- 각국 문자들을 정수로 mapping 하는 방법
  - 각 문자를 고유의 정수 값에 맵핑 (백만자 이상)
  - 21bit, 1~4byte 사용.
- BMP, SMP, surrogate 영역

- BMP 영역
  - UTF-16 에서 사용
    - 16bit 이내로 정의되는 영역 (Basic Multilingual Plane)
    - 자주 사용되는 문자들이 이 영역에 정의
  - SMP 영역
  - UTF-16 에서 사용
    - 17~21bit 영역 (Supplementary Multilingual Plane)
    - 드물게 사용되는 문자들이 이 영역에 정의
  - surrogate 영역
  - UTF-16 에서 사용
    - SMP 영역의 문자들을 인코딩 할 때 conflict 가 발생하지 않도록 하기 위해 BMP 영역에서 문자를 표현하지 않고 비어 있는 영역 (D800 ~ DFFF)
      - high surrogate D800 ~ DBFF
      - low surrogate DC00 ~ DFFF
    - UTF-16 에서만 사용
- UTF 인코딩 필요성 및 방식
  - 필요성
    - 유니코드 문자를 컴퓨터가 처리할 수 있는 바이트 형태로 변환
  - UTF-8 인코딩
    - 1byte, 1~7bit 영역

      0xxxxxxx 으로 표현
    - 2byte, 8~11bit 영역

      상위 5bit, 하위 6bit 을 나눠서 2byte 표현

110xxxxx 10xxxxxx

- 3byte, 12~16bit 영역, 한글이 여기에 포함

  4bit, 6bit, 6bit 으로 나눠서 3byte 표현

  1110xxxx 10xxxxxx 10xxxxxx

- 4byte, 17~21bit 영역

  3bit, 6bit, 6bit,6bit 으로 나눠서 4byte 표현

  11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

- UTF-16
  - 2 바이트: BMP영역

    BMP 영역은 변환 없이 2 바이트로 직접 표현
  - 4 바이트: SMP영역

코드에서 0x10000 을 빼서 20 비트의 값으로 20bit -> 각 상위, 하위 로 나눈다.

상위 110110yyyyyyyyyy 으로 표현 , 하위 110111xxxxxxxxxx 으로 표현

- UTF-32
  - 코드를 고정된 4byte 로 저장

3. 각자 자기 이름을 UTF-8로 인코딩
   - 금동환 → ea b8 88 eb 8f 99 ed 99 98
     - 금 → ea b8 88 → 11101010 10111000 10001000
     - 동 → eb 8f 99 → 11101011 10001111 10011001
     - 환 → ed 99 98 → 11101101 10011001 10011000