

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) . Int. Cl. ⁸ <i>G06F 3/023</i> (2006.01)		(45) 공고일자 (11) 등록번호 (24) 등록일자	2006년02월20일 10-0552085 2006년02월07일
(21) 출원번호 (22) 출원일자 번역문 제출일자 (86) 국제출원번호 국제출원일자	10-2000-7003211 2000년03월24일 2000년03월24일 PCT/US1998/020200 1998년09월24일	(65) 공개번호 (43) 공개일자 (87) 국제공개번호 국제공개일자	10-2001-0024309 2001년03월26일 WO 1999/15952 1999년04월01일
(81) 지정국	국내특허 : 알바니아, 아르메니아, 오스트리아, 오스트레일리아, 아제르바이잔, 보스니아 헤르체고비나, 바르바도스, 불가리아, 브라질, 벨라루스, 캐나다, 스위스, 중국, 쿠바, 체코, 독일, 덴마크, 에스토니아, 스페인, 핀란드, 영국, 그루지야, 가나, 감비아, 크로아티아, 헝가리, 인도네시아, 이스라엘, 아이슬랜드, 일본, 케냐, 키르키즈스탄, 북한, 대한민국, 카자흐스탄, 세인트루시아, 스리랑카, 리베이라, 레소토, 리투아니아, 룩셈부르크, 라트비아, 몰도바, 마다가스카르, 마케도니아공화국, 몽고, 말라위, 멕시코, 노르웨이, 뉴질랜드, 폴란드, 포르투칼, 루마ニア, 러시아, 수단, 스웨덴, 성가포르, 슬로베니아, 슬로바키아, 시에라리온, 타지키스탄, 투르크맨, 터키, 트리니아드토바고, 우크라이나, 우간다, 우즈베키스탄, 베트남, 세르비아 앤 몬테네그로, 짐바브웨,		
	AP ARIPO특허 : 가나, 감비아, 케냐, 레소토, 말라위, 수단, 스와질랜드, 우간다, 짐바브웨,		
	EA 유라시아특허 : 아르메니아, 아제르바이잔, 벨라루스, 키르키즈스탄, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크맨,		
	EP 유럽특허 : 오스트리아, 벨기에, 스위스, 사이프러스, 독일, 덴마크, 스페인, 핀란드, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투칼, 스웨덴,		
	OA OAPI특허 : 부르키나파소, 베닌, 중앙아프리카, 콩고, 코트디브وار, 카메룬, 가봉, 기니, 기니 비사우, 말리, 모리타니, 니제르, 세네갈, 차드, 토고,		
(30) 우선권주장	60060223 09160433	1997년09월25일 1998년09월24일	미국(US) 미국(US)
(73) 특허권자	테직 커뮤니케이션 인코포레이티드 미국, 98121 워싱턴, 시애틀, 슈이트 250, 웨스턴 애비뉴 2001		
(72) 발명자	플린 챔에드워드피. 미국워싱턴98119시애틀웨스트키너플레이스118		
	그루버 데일엘. 미국미시간48915렌싱웨스트제네시731		
	그런벽체릴 미국워싱턴98070바순피.오.박스1829		
	킹마틴티.		

미국위성턴98070바순피.오.박스1829

쿠슬러클리포드에이.

미국위성턴98070바순포인트바손드라이브사우스웨스트10857

(74) 대리인

박종혁
정진상
장용식

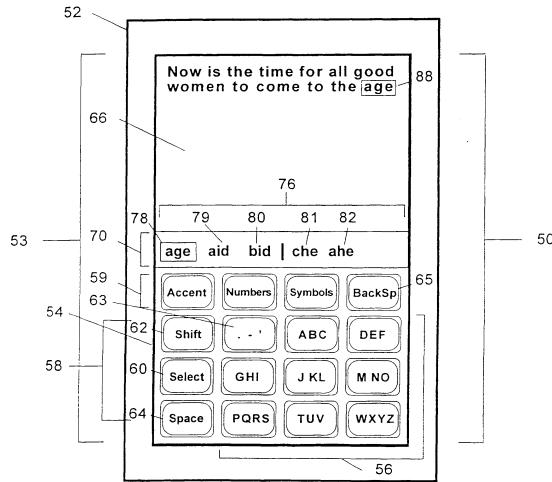
심사관 : 김기천

(54) 감소된 키보드 명확화 시스템

요약

감소된 키보드 명확화 시스템. 본 키보드는 12에서 16개의 키를 가지고 있고, 숫자와 다른 기호로 라벨된 9개의 키, 및 9개 + 1이상의 키는 10 디지트중 하나와 각각 관련되어 있다. 텍스트 입력 키스트로크는 모호하다. 사용자는 동일 키 스트로크와 관련된 복수의 워드중 하나와 일치할 수 있는 키 스트로크 시퀀스를 제한하면서, 각 워드의 끝에서 모호하지 않은 문자 해석을 가진 키 또는 제한된 "Select" 키를 친다. 각각의 입력 키 스트로크 시퀀스는 완성된 어휘 모듈, 그리고 사용 빈도수를 줄이는 순서로 사용자에게 주어진 키 스트로크 시퀀스를 일치시키는 워드로 처리된다. 상기 어휘는 추가 처리되지 않는 상당히 데이터 압축을 지원하는 특정 형태로 저장된다. 추가로, 첫문자가 키 스트로크와 일치하는 긴 워드의 스템은 사용 빈도수를 줄이는 순서로 사용자에게 또한 제공된다. 스템과 관련된 워드의 빈도수가 상당히 높으면, 그 스템은 디폴트 워드 객체로서 디스플레이된다. 복수의 구두점 문자와 관련된 모호한 키의 동작은 주위 키 스트로크의 항목으로부터 명확하게 된다.

대표도



색인어

어휘 모듈, 워드 객체, 키 스트로크, 키보드, 사용 빈도수, 문자 해석

명세서

기술분야

본 발명은 일반적으로 감소된 키보드 시스템에 관한 것이고, 보다 상세하게는 모호한 키스트로크를 해결하기 위해 워드-레벨 명확화를 이용하는 감소된 키보드 시스템에 관한 것이다.

배경기술

여러 해동안, 휴대용 컴퓨터는 점점 소형화되었다. 보다 소형의 휴대용 컴퓨터를 생산하는데 영향을 주는 주요 크기제한 부품은 키보드였다. 규격의 타이프라이터 사이즈의 키가 사용되면, 휴대용 컴퓨터는 적어도 키보드만큼의 크기를 가져야 한다. 감소된 키보드가 휴대용 컴퓨터에 사용되었지만, 그 감소된 키보드 키는 너무 적어서 사용자가 쉽고 빠르게 조작하는 것이 불가능하다는 것을 알았다.

휴대용 컴퓨터에 풀 사이즈의 키보드를 장착하면, 그 컴퓨터의 진정한 휴대용도를 또한 방해한다. 대부분의 휴대용 컴퓨터는, 사용자가 양손으로 타이핑할 수 있는 평면 작업대 위에 그 컴퓨터를 놓지 않고는 작동시킬 수 없다. 사용자는 서있고 움직이는 동안에 휴대용 컴퓨터를 쉽게 사용할 수 없다. 가장 최근 세대의 소형 휴대용 컴퓨터, 즉, 개인 정보 이동단말기 (PDA)로, 대부분의 회사들은 PDA에 수기 인식 소프트웨어를 내장시킴으로써, 이러한 문제점에 대처하려구 노력하고 있다. 사용자는 터치 감지 패널 또는 스크린상에 기록함으로써 텍스트를 바로 입력할 수 있다. 이러한 수기 텍스트는 인식 소프트웨어에 의해 디지털 데이터로 변환된다. 불행히도, 펜으로 활자체를 쓰고 기록하는 것이 일반적으로 타이핑보다 느리다는 사실에 추가로, 지금까지는 수기 인식 소프트웨어의 정확성과 속도가 만족스럽지 않다. 더욱이, 텍스트 입력을 필요로 하는 오늘날의 소형 컴퓨팅 장치는 여전히 소형화되고 있다. 양방향 페이징, 셀룰러 전화, 및 다른 휴대용 무선 기술의 발달로, 소형의 휴대가능 양방향 페이징 시스템과, 전자우편("E-mail")을 모두 송수신 할 수 있는 시스템이 요구되고 있다.

그러므로, 사용자가 한 손으로 컴퓨터를 잡은 상태에서 다른 한 손으로 동작가능하고 소형인 컴퓨터에 텍스트를 입력하기 위한 키보드를 개발하는 것이 바람직할 수 있다. 앞선 개발 작업은 감소된 수의 키를 가진 키보드를 이용하는 것을 고려하였다. 터치 톤 전화의 키패드 배치에 의해 알 수 있는 바와 같이, 대부분의 축소형 키보드는 3×4 배열의 키를 사용하였다. 이러한 배열의 각각의 키는 복수의 문자를 포함하고 있다. 그러므로, 각각의 키스트로크는 몇몇 문자중 하나를 지시하고 있기 때문에, 사용자가 일련의 키를 입력할 때 모호하다. 키스트로크 시퀀스의 모호성을 해결하기 위한 몇가지 해결책이 제안되었다.

축소형 키보드상에 입력된 문자를 명확하게 표시하는 한 가지 제안된 해결책은 각각의 문자를 표시하기 위해 사용자가 둘 이상의 키스트로크를 입력하여야 한다. 이러한 키스트로크는 동시에(코딩) 또는 순서로(다중 스트로크 표시법) 입력될 수 있다. 코딩 또는 다중 스트로크 표시법중 어느 하나도 사용상 간편하고 효율성이 있는 키보드를 생산하지 못하였다. 다중 스트로크 표시법은 불충분하고, 코딩은 사용하고 배우는 것이 복잡하다.

모호한 키스트로크 시퀀스에 대응하는 정확한 문자 시퀀스를 결정하는 다른 제안된 해결책은 John L.Arnott 와 Muhammad Y.Javad의, 부가 및 대체 통신에 대한 국제 협회의 저널에 발행된 "Probabilistic Character Disambiguation for Reduced Keyboards Using Small Text Samples" 문헌(이하 "아노트 문헌")에 요약되어 있다. 아노트 문헌은 대부분의 명확화법이 주어진 문맥에서의 문자의 모호성을 해결하기 위해 관련 언어의 문자 시퀀스의 공지된 통계를 사용하고 있다고 기재하고 있다. 즉, 현존하는 명확화 시스템은 키스트로크의 적절한 해석을 결정하기 위해 사용자에 의해 입력되는 것으로서, 모호한 키스트로크를 통계적으로 분석한다. 아노트 문헌은 일부 명확화 시스템은 축소된 키보드로부터의 텍스트를 디코딩하기 위해 워드-레벨 명확화법을 이용하고자 시도하였음을 개시하고 있다. 워드-레벨 명확화은 수신된 키스트로크의 시퀀스를 워드의 끝을 표시하는 명확한 문자를 수신한 후 사전내의 가능한 매칭과 비교함으로써 전체 워드를 명확하게 한다. 아노트 문헌은 워드-레벨 명확화법의 다수의 단점을 개시하고 있다. 예를 들어, 워드-레벨 명확화법은 비일상적인 워드를 표시할 때의 한계와 사전에 포함되어 있지 않은 워드의 디코딩 불가능으로 인해, 워드를 정확하게 디코딩하는데 종종 실패한다. 디코딩의 한계로 인해, 워드-레벨 명확화법은 문자당 하나의 키스트로크의 효율성을 가진 자연스러운 영어 텍스트의 무에러 디코딩을 부여하지 않는다. 그러므로, 아노트 문헌은 워드-레벨 명확화법보다는 오히려 문자 레벨 명확화법에 초점을 두고 있고, 문자 레벨 명확화법은 대다수의 유망한 명확화 기술로 보인다고 개시하고 있다.

워드-레벨 명확화법에 근거한 하나의 제안된 해결책이, 1982년에 아카데미 출판사에 의해 발행되고 I.H.Witten에 의해 저술된 Principles of Computer Speech(이하, "위튼 접근법")에 개시되어 있다. 위튼은 전화 터치 패드를 이용하여 입력된 텍스트로부터의 모호성을 감소시키는 시스템을 개시하고 있다. 위튼은 24,500 워드 사전내의 대략 92%의 워드에 대하여, 키스트로크 시퀀스를 사전과 비교할 때 모호성이 발생하지 않는다고 개시하고 있다. 그러나, 모호성이 발생하는 경우에, 위튼은, 사용자에게 모호성을 표현하고 사용자가 모호한 엔트리중에서 선택하게 하는 시스템에 의해 모호성은 상호 해결되어야 한다고 개시하고 있다. 그러므로, 사용자는 각 워드의 후미에서 그 시스템의 예측에 응답하여야 한다. 이러한 응답은 시스템의 효율성을 저하시키고, 주어진 텍스트 세그먼트를 입력하는데 필요한 키스트로크의 수를 증가시킨다.

휴대용 컴퓨터에 사용하기 적합한 감소된 명확화 키보드에 대한 필요한 효율성을 얻기 위해 텍스트의 세그먼트를 입력하는데 필요한 키스트로크 수를 감소시키려고 상당히 노력하였다. 그 사용이 가장 유용한 하드웨어 플랫폼의 종류에 위드 레벨 명확화를 성공적으로 구현하려고 노력하고 있다. 상기된 바와 같이, 이러한 디바이스는 양방향 페이저, 셀룰러 전화, 및 다른 휴대용 무선 통신 장치를 포함한다. 이러한 시스템은 배터리식 전원이고, 결과적으로 하드웨어 설계와 자원 이용면에서 가능한 한 절약할 수 있도록 설계되어 있다. 이러한 시스템에 따라 동작하도록 설계된 애플리케이션은 프로세서의 대역 이용도와 메모리 필요성 모두를 최소화시켜야 한다. 이러한 두 가지 요인은 일반적으로 역상관 관계에 있다. 위드-레벨 명확화 시스템은 작용하는 워드의 상당양의 데이터베이스를 필요로 하고 만족스러운 사용자 인터페이스를 제공하기 위해 입력 스트로크에 신속하게 대응하여야 하기 때문에, 이것을 이용하는데 필요한 프로세싱 시간에 상당히 영향을 주지 않고 필요한 데이터베이스를 압축할 수 있는 것이 상당히 유리할 것이다.

워드-레벨 명확화의 어떤 애플리케이션에서 접할 수 있는 또 다른 문제점은 입력되는 키스트로크에 대하여 사용자에게 충분한 피드백을 제공하는 것이다. 보통의 타이프라이터 또는 워드 프로세서로, 각각의 스트로크는 입력되자마자 사용자에게 디스플레이될 수 있는 독특한 문자를 표현한다. 그러나, 워드-레벨 명확화법으로는, 각각의 키스트로크가 다중 문자를 표현하고, 키스트로크의 임의의 시퀀스가 다중 워드 또는 워드 스템(stem)에 일치할 수 있기 때문에, 이것은 종종 가능하지 않다. 특히 이것은 사용자가 완료 키 시퀀스가 입력된 때까지 에러가 일어났음을 확인할 수 없어 소망 워드가 나타나지 못하기 때문에, 스펠링 또는 키스트로크 에러를 일으킬 때 문제가 된다.

상기 노력에 중점을 둔 워드-레벨 명확화 시스템은 국제 협력 조약 특허 출원 PCT/US96/12291에 개시되어 있고, "Reduced Keyboard Disambiguating System", 국제 공보 WO 97/05541(이하, "WO 97/05541 공보")로서 2월 13일에 공개되어 있다.

WO 97/05541 공보에 개시된 시스템은: 복수의 문자와 기호가 적어도 몇개의 키에 할당된 키보드; 하나이상의 어휘 모듈을 저장하는 메모리; 시스템 원문 출력이 제공되는 디스플레이; 및 상기 키보드, 메모리, 및 디스플레이에 동작적으로 연결되어 있는 프로세서;를 포함하고 있다. 워드 및 워드 스템(워드의 시작부에 대응하는 문자 시퀀스)은 복수의 상호 접속 노드를 포함하는 트리 구조를 이용하여 어휘 모듈에 저장되어 있다.

WO 97/05541 공보에 개시된 워드 레벨 명확화 시스템의 동작시에, 사용자가 키스트로크 시퀀스를 입력할 때, 시스템 프로세서는 입력된 키 스트로크 시퀀스에 대응하는 워드 및 워드 스템을 구성하기 위해 시스템 메모리에 액세스한다. 특히, 각각의 입력된 키 스트로크 시퀀스는, 교대로 워드 및 워드 스템에 대응하는 어휘 모듈 트리 구조의 노드에 대응한다. 추가 키 스트로크가 입력될 때, 시스템 프로세서는 진행한 키 스트로크 시퀀스(즉, 최종 키 스트로크에 포함되어 있지 않은 키 스트로크 시퀀스)와 관련된 워드 및/또는 워드 스템 세트를 결합함으로써 하나이상의 새로운 워드 스템 및/또는 워드 세트를 형성한다. 메모리 트리 구조와 관련하여, 하나이상의 워드 스템 및/또는 후보 워드 세트는 앞선 키 스트로크 시퀀스를 표현하고, 특히 총 키 스트로크 시퀀스에 대응하는 모든 가능한 워드 스템 및 워드를 표현하는 노드와 상호 접속되는 트리 구조의 노드와 관련된 워드 및/또는 워드 스템에 액세스함으로써 생성된다.

WO 97/05541 공보에 개시된 장치에 따라서, 워드 및 워드 스템은 디스플레이되는 워드 및 워드 스템과 관련된 사용 빈도 수에 따라서 시스템 디스플레이상에 표현된다. 후보 워드 리스트가 디스플레이될 때, 본 시스템의 사용자는 소망의 후보 워드를 선택하기 위해 지정된 키보드 키를 동작시킨다.

삭제

발명의 상세한 설명

본 발명은 키스트로크의 모호성을 해결하기 위해 워드-레벨 명확화법을 이용하는 감소된 키보드를 제공한다. 일실시예에서, 본 시스템은 터치 감지식 디스플레이 패널을 포함하고 있고, 디스플레이의 표면과의 접촉부는 접촉 위치에 대응하여 본 시스템에 입력 신호를 발생시킨다. 대안으로, 상기 키보드는 폴 사이즈의 메카니즘 키로 구성될 수 있다.

바람직한 일실시예에서, 9개의 기호와 문자 키가 3 내지 6개의 추가 특수 기능키와 함께 3×3 배열로 그룹화되어 있다. 복수의 문자와 기호는 키의 일부에 할당되어 있어서, 이러한 키(이하 "데이터 키")의 키 스트로크는 모호하다. 사용자는 각각의 키 스트로크가 하나의 워드의 하나의 문자 입력에 대응하는 키 스트로크 시퀀스를 입력할 수 있다. 개별 키 스트로크가 모호하기 때문에, 키 스트로크 시퀀스는 하나 이상의 워드를 동일 수의 문자와 매칭시킬 수 있다. 키 스트로크 시퀀스는 그 시퀀스를 대응하는 저장 워드에 매칭시키는 어휘 모듈 또는 다른 해석에 의해 처리된다. 키스트로크 시퀀스와 일치하는 워드 또는 워드 스템은 각각의 키 스트로크가 수신될 때 디스플레이상의 선택 리스트로 사용자에게 제공된다.

본원발명의 일태양에 따라, 키 중 하나는 복수의 구두점 문자에 할당된다. 바람직한 일실시예에 있어서, 마침표 '.', 하이픈 '-' 및 어포스트로피 "'"는 키 중 하나에 할당된다. 사용자는 구두점 문자가 소망되는 포인트에서 키의 단일 활성화에 의해

키의 문자 중 하나를 타이핑할 수 있다. 복수의 구두점 문자가 키에 할당되기 때문에, 키스트로크는 모호하다. 본원발명의 시스템은 구두점 문자가 의도된 이전 및 이후 키스트로크로부터 결정하여 그것을 자동으로 발생시킨다. 또한 키스트로크 시퀀스의 대안의 해석은 선택 리스트로 사용자에게 제공될 수 있다.

바람직하게, 가장 일반적으로 사용되는 워드가 먼저 제공된 상태에서, 사용하는 빈도수를 감소시키기 위해 워드 해석법이 주어진다. 선택 리스트내의 항목은 한 번이상 선택 키를 누름으로써 선택된다. 키 스트로크는 백스페이스 키를 누름으로써 "undone"될 수 있다.

사용자는 입력된 키스트로크 시퀀스를 한정하기 위해 선택 키를 누른다. 선택 키를 수신한 후, 명확화 시스템은 가장 자주 사용되는 빈도수를 선택하고 구성된 문장에 그 워드를 부가한다. 선택 키는 입력된 키 스트로크 시퀀스를 한정하는데 사용된다. 개별적으로 명시된 스페이스 키는 스페이스 문자를 생성하는데 사용되고, 또한 입력된 키 스트로크 시퀀스를 한정하는 작용을 한다. 다른 바람직한 실시예에서, 선택 키는 선택된 키에 수반되는 스페이스를 생성하는 점에서 제 2 기능으로 "오버로드"된다. 즉, 감소된 키보드 명확화 시스템은 자동적으로 워드간에 적절한 거리를 삽입한다.

선택 키는 사용자에 의해 표현된 선택 리스트부터 보다 적게 사용되는 워드를 선택하는데 또한 사용된다. 선택 리스트의 최상부에 사용자에게 주어진 워드가 바라는 워드가 아니라면, 사용자는 가장 자주 사용되는 워드로부터 두번째로 자주 사용되는 워드까지 선택 키를 나아가고, 다시 3번째로 자주 사용되는 워드 등으로 나아가기 위해 선택 키를 누른다. 감소된 키보드 구조의 이러한 실시예에는 한번 선택되었을 때 하나의 엔트리를 동작시키는 전용 "실행" 또는 "수용" 키를 가지고 있다. 소망의 워드가 사용자에 의해 선택되면, 출력을 위해 자동적으로 "수용"되고, 다음 기호 또는 문자 키 스트로크를 수신하는 즉시 구성되어 있는 문자에 부가된다.

현재 키 스트로크 시퀀스에 대응하는 노드와 관련된 가장 빈번한 워드는 아니지만, 총 빈도수가 현재 노드와 관련된 가장 빈번한 워드의 총 빈도수보다 큰 보다 긴 워드 스템 또는 복수의 워드를 형성하는 워드 또는 워드 스템 객체를 제 1 객체로서 본 시스템이 제공할 수 있게 하는 정보를, 키 스트로크를 명확하게 하는데 사용되는 언어 데이터베이스가 포함하고 있다. 이러한 "스템 측진" 특징은, 사용자가 삽입점에 디스플레이된 워드를 볼 수 있는 곳에서, 선택 리스트가 스크린상에 실질적으로 디스플레이되지 않도록 제한된 디스플레이 영역을 가진 시스템에 특히 유용하고, 선택 리스트로부터의 제 1 객체이다. 그 결과는, 객체 전체적으로 키스트로크간에 보다 적은 수로 변경하려고 하고, 적은 "점프"를 표현하고, 결과적으로 인터페이스를 적게 트러지게 하고 적게 혼동스럽게 한다.

워드 및 워드 스템과 같은 객체는 트리 구조를 이용한 하나이상의 어휘 모듈에 저장되어 있다. 이러한 구조에서, 특정 키 스트로크 시퀀스에 대응하는 워드는 바로 전의 키 스트로크 시퀀스(즉, 최종 키 스트로크가 없는 특정 키 스트로크 시퀀스)와 관련된 워드 및 워드 스템 세트를 이용하여 구성되어 있다. 이러한 방식으로 워드를 구성하면, 워드 스템은 트리 구조의 최상부에 단 한번 저장되고, 그들로부터 구성된 모든 워드에 의해 공유되기 때문에, 어휘 모듈의 저장 공간을 감소시킨다. 트리 구조는 또한 저장된 객체를 위치시키는데 조사가 필요하지 않기 때문에, 처리할 필요 조건을 상당히 감소시킨다. 트리 데이터 구조에 저장된 워드 및 워드 스템은 어떤 아이템이 사용자에게 먼저 디스플레이되어야 하는지를 표시하는 다른 랭킹 정보 또는 빈도수를 포함할 것이고, 추가로 처리할 필요조건을 감소시킨다. 본 발명의 실행에서, 트리 데이터 구조는, 키 스트로크 시퀀스와 관련된 객체를 검색하는데 이용될 때 추가 처리 부담을 야기하지 않고 데이터베이스에 필요한 총 사이즈를 추가로 압축한다. 수정된 트리 구조의 다른 측면은 키 스트로크 시퀀스에 객체를 연관시키는 규정의 자동 식별이다. 이러한 방식으로, 어휘 모듈은 생성할 때 초기에 사용되지 않는 워드 및 워드 스템 객체에 키스트로크 시퀀스를 성공 가능성이 높게 연관시킨다.

삭제

삭제

삭제

바람직한 실시예에서의 키의 내부적 논리적 표현법은 실제 키상의 라벨에 의해 표현되는 물리적인 배열을 반영할 필요는 없다. 예를 들어, 프랑스 어휘 모듈을 표현하기 위해 구성된 데이터베이스에서, 3개의 추가 기호(ÀÇÀ)는 문자의 악센트가 없는 버전과 관련된 키(ABC)와 또한 관련될 수 있다. 이로써, 사용자는, 관련 악센트 문자에 대해 논리적으로 관련된 물리적 키를 단순히 동작시킴으로써, 문자당 하나의 키 동작만을 수행하여, 특수 악센트 문자를 포함하는 워드를 리콜하고 타이핑할 수 있다.

다중 문자를 키에 할당하는 결합 효과, 선택 키를 사용한 워드의 한정, 선택 리스트내의 제 1 워드로서 가장 일반적으로 발생하는 워드 또는 워드 스템의 표현, 선택 리스트내의 다중 해석의 포함, 수반하는 워드의 제 1 키 스트로크에 의한 문장에의 선택된 워드의 자동 추가, 스페이스의 자동 추가, 상당한 처리 부담없이 명확하게 상당한 데이터베이스를 압축할 수 있는 능력, 문자의 비악센트 버전과 관련된 키를 타이핑함으로써 삽입된 특수 악센트 문자를 가진 워드를 생성할 수 있는 능력, 및 키 스트로크의 문맥을 기초로 하여 단일 키에 할당된 복수의 구두점 문자중에서 자동적으로 명확하게 할 수 있는 능

력은 월등한 결과를 만든다: 다수의 언어에 대하여, 원문의 표현 자료에서 발견되는 99%이상의 언어가 상당히 높은 효율성을 가진 시스템에 의해 타이핑될 수 있다. 이러한 언어의 대략 95%에 있어서, 종래의 키보드로 입력한 워드에 대하여와 같이, 감소된 키 명확화 시스템으로 워드를 입력하는데 동일 수의 키 스트로크가 필요하다. 워드가 악센트 문자를 포함하는 경우에, 종래의 키보드로 보다는 보다 적은 수의 키 스트로크로 워드 입력이 종종 이루어질 수 있다. 워드가 사용 오더의 빈도수로 표현될 때, 원하는 워드는 가장 자주 주어지는 제 1 워드이고, 단지 자주 주어지는 워드이다. 사용자는 일반적인 수 이하의 키 스트로크로 다음 워드를 입력할 수 있다. 따라서, 소수의 풀사이즈 키를 가진 키보드를 이용하여 텍스트의 고속 입력이 이루어진다.

여기서 개시된 감소된 키보드 명확화 시스템은 본 시스템을 포함하는 다른 장치 또는 컴퓨터의 사이즈를 감소시킨다. 감소된 수의 키로 디바이스는 한손으로 잡고 있고 다른 한손으로 동작될 수 있도록 구성될 수 있다. 상기 시스템은 정확한 고속 텍스트 입력으로부터 이로운 다른 소형의 전자 디바이스, 양방향 페이저, 또는 PDA에 이용하는 것이 특히 바람직하다. 본 시스템은 압축된 데이터베이스를 이용할 때 추가 처리 대역을 필요로 하지 않고 키 스트로크 시퀀스를 명확하게 하는 상당량의 데이터베이스를 효율적으로 압축한다. 본 시스템은 제한된 디스플레이 스크린 영역을 또한 가질 수 있는 제한된 수의 메카니즘 키를 가진 디바이스 또는 터치스크린 계통의 디바이스상에 구현될 때 효율적이고 간략하게 할 수 있다.

도면의 간단한 설명

본 발명의 전술한 측면과 다수의 수반 잇점은 동일한 것이 첨부한 도면과 결부시킬 때, 다음의 상세한 설명을 참조하여 보다 충분히 이해되는 것처럼 보다 쉽게 알 수 있게 될 것이다.

도 1은 본 발명의 감소된 키보드 명확화 시스템을 내장한 휴대용 컴퓨터의 바람직한 실시예의 개략도;

도 2는 도 1의 감소된 키보드 명확화 시스템의 하드웨어 블록도;

도 3A-3C은 감소된 키보드 명확화 시스템용 소프트웨어를 명확하게 하는 바람직한 실시예의 흐름도를 전체적으로 도시하는 도면;

도 4는 어떤 텍스트 객체가 감소된 명확화 시스템용 트리 데이터 구조의 각각의 노드에서 디폴트 객체로서 사용되어야 하는지를 결정하는 소프트웨어의 바람직한 실시예의 흐름도;

도 5A-5D는 감소된 키보드 명확화 시스템의 키와 관련된 복수의 구두점 문자중 의도하고자 하는 구두점 문자를 명확하게 하는 각각의 키 스트로크로 수행되는 소프트웨어 프로세스의 바람직한 실시예의 흐름도; 도 5A-5D는 키 스트로크의 초기 수신 즉시 수행되는 프로세싱, 및 도 5D는 키스트로크의 모든 다른 프로세싱이 완료되었음을 수행하는 프로세싱;

도 6은 물리적인 키상에 표시하는 문자의 추가 악센트 변수를 포함한 논리적 연관성의 예와 키로의 기호의 물리적 연관성을 비교하는 단계;

도 7은 논리 기호를 키 색인과 연관시킨 테이블의 예를 도시하는 도면;

도 8A는 어휘 모듈 트리의 노드로 바람직한 내부 배열을 도시하는 도면;

도 8B는 명령의 바람직한 실시예의 어의 구성요소를 도시하는 도면;

도 9는 바람직한 실시예의 노드 구조로 가능한 내부 데이터 아이템의 4개의 예를 도시하는 도면;

도 10은 압축해제된 어휘 모듈의 바람직한 트리 구조를 도시하는 도면;

도 11은 어휘 모듈로부터 검색되는 프로세스에서 객체의 중간 저장에 대한 바람직한 실시예인 객체 리스트의 예시적인 상태를 도시하는 도면;

도 12는 키 리스트를 누르면 어휘 모듈로부터 텍스트 객체를 검색하는 소프트웨어 프로세스의 바람직한 실시예의 흐름도;

도 13은 단일 키를 누르면 어휘 모듈의 트리 구조를 좌우로 이동하고 객체 리스트의 상태를 변경시키는 소프트웨어 프로세스의 바람직한 실시예의 흐름도;

도 14는 폴드식 압축 어휘 모듈을 만드는 소프트웨어 프로세스의 바람직한 실시예의 흐름도;

도 15는 어휘 모듈의 트리 데이터 구조를 폴드하는 소프트웨어 프로세스의 바람직한 실시예의 흐름도;

도 16은 주어진 노드와 비교하여 상당한 중복성을 가진 어휘 모듈의 트리에 제 2 노드를 위치시키는 소프트웨어 프로세스의 바람직한 실시예의 흐름도;

도 17A 및 17B은 어휘 모듈로 트리내의 두 개의 노드사이의 중복성을 연산하는 소프트웨어 프로세스의 바람직한 실시예의 흐름도를 전체적으로 도시하는 도면;

도 18A 내지 도 18K는 대표적인 사용시 감소된 키보드 명확화 시스템의 바람직한 실시예의 개략도.

실시예

바람직한 실시예의 상세한 설명

I . 시스템 구조와 주요 동작

도 1을 참조하면, 본 발명에 따른 감소된 키보드 명확화 시스템(50)은 팜탑 휴대용 컴퓨터(52)에 내장되어 설명되어 있다. 휴대용 컴퓨터(52)는 터치스크린 디스플레이(53)상에 구현된 감소된 키보드(54)를 포함하고 있다. 본 출원의 목적을 위해, 용어 "키보드"는 키, 이산 메카니즘 키, 멤브레인 키 등에 대해 한정된 영역을 가진 터치 스크린을 포함한 한정된 영역을 가진 임의의 입력 디바이스를 포함하는 것으로 광범위하게 정의된다. 키보드(54)는 표준 QWERTY 키보드로부터의 감소된 수의 데이터 입력 키를 가지고 있다. 바람직한 일실시예에서, 키보드는 4×4 로 배열된 16개의 표준 풀 사이즈 키를 포함하고 있다. 보다 상세하게는, 바람직한 키보드는 구두점 키(63)를 포함한 3×3 어레이로 배열된 9개의 데이터 키(56), 선택 키(60), 시프트 키(62), 및 스페이스 키(64)를 포함한 좌측 열의 3개의 시스템 키(58), 및 이전 키 스트로크를 삭제하는 상위 행(59)의 백스페이스 키(65)를 포함하고 있다. 시스템 키의 상위 행(59)은 악센트 문자, 숫자, 및 기호 각각을 타이핑하는 모드로 진입하기 위해 Accent, Numbers, 및 Symbols로 라벨 표시된 3개의 MODE 키를 또한 포함하고 있다.

키보드(54)의 각각의 키상의 문자의 바람직한 실시예는 도 1에 도시되어 있다. 도 1은 영문용 키보드(54)의 모호한 데이터 키(63)와 관련된 구두점 문자의 바람직한 실시예를 또한 도시하고 있다.

데이터는 감소된 키보드(54)의 키스트로크를 통해 명확화 시스템에 입력된다. 사용자가 그 키보드를 이용하여 키 스트로크 시퀀스를 입력할 때, 텍스트는 컴퓨터 디스플레이(53)상에 디스플레이된다. 두 개의 영역이 사용자에게 정보를 디스플레이하기 위해 디스플레이상에 정의되어 있다. 상위 텍스트 영역(66)은 사용자에 의해 입력된 텍스트를 디스플레이하고, 편집 및 텍스트 입력용 버퍼로서 작용한다. 텍스트 영역아래에 위치한 선택 리스트 영역(70)은 사용자에 의해 입력된 키 스트로크 시퀀스에 대응하는 다른 해석 및 워드 리스트를 제공한다. 아래에 추가로 상세히 설명되는 바와 같이, 선택 리스트 영역(70)은 사용자가 입력된 키스트로크의 모호성을 해결할 수 있게 한다. 다른 바람직한 실시예에서, 본 시스템은 제한된 디스플레이 스페이스를 가진 디바이스상에 구현될 수 있고, 생성된 텍스트내에 삽입점(88)에서 현재 선택된 또는 가장 유사한 워드 객체만을 디스플레이한다.

감소된 키보드 명확화 시스템 하드웨어의 블록도가 도 2에 제공되어 있다. 키보드(54)와 디스플레이(53)는 적절한 인터페이스 회로를 통해 프로세서(100)에 접속되어 있다. 선택사양으로, 스피커(102)가 또한 프로세서에 접속된다. 프로세서(100)는 키보드로부터 입력을 수신하고, 디스플레이 및 스피커로의 모든 출력을 관리한다. 프로세서(100)는 메모리에 접속되어 있다. 메모리는 랜덤 액세스 메모리(램)와 같은 임시 기억 매체 및, 판독 전용 메모리(롬), 플로피 디스크, 하드 디스크, 및 CD-ROM과 같은 영구 기억 매체의 결합체를 포함한다. 메모리(104)는 시스템 동작을 통제하는 모든 소프트웨어 루틴을 포함하고 있다. 바람직하게, 메모리는 운영 시스템(106), 명확화 소프트웨어(108), 및 아래에 상세히 설명될 관련 어휘 모듈(110)을 포함하고 있다. 선택사양으로, 메모리는 하나이상의 응용 프로그램(112, 114)을 포함할 수 있다. 응용 프로그램의 예는 워드 프로세서, 소프트웨어 사전, 및 외국어 번역기를 포함하고 있다. 언어 분석 소프트웨어는 또한 응용 프로그램으로서 제공될 수 있고, 감소된 키보드 명확화 시스템이 통신 목적으로 사용되게 한다.

도 1을 참조하면, 감소된 키보드 명확화 시스템(50)으로, 사용자는 한손으로 텍스트 또는 다른 데이터를 신속하게 입력할 수 있다. 데이터는 데이터 키(56)를 이용하여 입력된다. 데이터 키 각각은 복수의 문자, 숫자, 및 다른 기호에 의해 키의 최상부에 표현되는 복수의 의미를 가지고 있다. (이러한 설명을 위해, 각각의 데이터 키는 우상부의 데이터 키를 식별하기 위

해 데이터 키의 중앙 행의 기호, 예 DEF에 의해 식별될 수 있다.) 개별 키가 복수의 의미를 가지고 있기 때문에, 키 스트로크 시퀀스는 그 의미에 대해서는 모호하다. 사용자가 데이터를 입력할 때, 여러 키스트로크의 해석은, 사용자가 어떤 모호성을 해결하는데 도움이 되도록, 디스플레이상의 복수의 영역에 디스플레이된다. 이용가능한 충분한 디스플레이 영역을 가진 디스플레이에서, 입력된 키스트로크의 가능한 해석의 선택 리스트(76)는 선택 리스트 영역(70)으로 사용자에게 제공된다. 선택 리스트의 제 1 입력(78)은 디폴트 해석으로서 선택되고, 삽입점(88)에서 텍스트 영역(66)에 디스플레이된다. 바람직한 실시예에서, 이러한 입력은 선택 리스트(76)와 삽입점(88) 모두에서 주위에 그려진 실선 박스로 디스플레이된다. 포맷 동작은 삽입점 객체와 선택 리스트사이의 가시적인 관계를 설정하고, 이러한 객체가 현재의 선택 리스트에서의 가장 자주 발생하는 객체인 이유로 절대적으로 선택된다. 대안으로, 제공된 선택 리스트가 없고, 디폴트 객체만(선택 키의 임의의 동작에 앞서 선택 리스트에 먼저 디스플레이될 수 있는 객체), 또는 하나가 명백하게 선택되었다면 현재 선택된 객체는 삽입점 (88)에서 디스플레이된다.

입력된 키 스트로크의 가능한 해석의 선택 리스트(76)는 다수의 방법으로 정렬될 수 있다. 동작의 정상적인 모드에서, 키 스트로크는 워드를 스펠링하는 문자의 엔트리로서 초기에 해석(이하, "워드 해석")된다. 선택 리스트내의 엔트리(78, 79, 80)는, 키 스트로크 시퀀스에 대응하는 가장 일반적인 워드가 먼저 리스트되도록 엔트리가 정렬된 상태에서, 입력된 키 스트로크 시퀀스에 대등하는 워드이다. 예를 들어, 도 1에 도시된 바와 같이, 키 스트로크 시퀀스 ABC, GHI, 및 DEF는 사용자에 의해 입력된다. 키가 입력될 때, 어휘 모듈 조사는 키스트로크 시퀀스를 일치시키는 워드를 배치하도록 동시에 수행된다. 어휘 모듈로부터 식별된 워드는 선택 리스트(76)로 사용자에게 디스플레이된다. 워드는 가장 일반적으로 사용되는 워드가 먼저 리스트되는 상태에서 사용빈도수에 따라서 분류된다. 예시적인 키스트로크 시퀀스를 사용하면, 워드 "age", "aid", 및 "bid"는 키스트로크 시퀀스에 대응하는 가장 사실적인 워드인 것으로서 어휘 모듈로부터 식별된다. 3개의 식별된 워드중에서, "age"는 "aid" 또는 "bid"보다 자주 사용되고, 그래서 선택 리스트에서 먼저 리스트되어 있다. 제 1 워드는 디폴트 해석으로서 간주되고 삽입점(88)에서 텍스트로서 임시 배치된다. 선택 키를 처리하기 전에, 디폴트 해석으로서 간주된 이러한 제 1 워드는 삽입점(88)과 식별 포맷동작을 이용하여 선택 리스트(76)에 배치된다. 예를 들어, 도 1에서와 같이, 워드는 워드를 포함하기에 충분히 큰 실선으로 그려진 박스내에 텍스트로서 나타난다. 스크린상에 실제 선택 리스트를 도시하기에 충분한 디스플레이가 없는 시스템에서, 잠정적으로 일치하는 워드의 리스트는 메모리에 저장되고, 일치하는 텍스트 객체의 상대적인 빈도수에 따라서 분류된다.

소망하는 워드에 대응하는 키 스트로크 시퀀스의 엔트리를 수반하는 바람직한 실시예에서, 사용자는 스페이스 키(64)를 단순히 누른다. 디폴트 워드(선택 리스트내의 제 1 워드)는 삽입점에서 직접 출력되고, 선택 리스트는 제거되고, 스페이스는 삽입점(88)에서 텍스트 영역으로 바로 출력된다. 대안으로, 모호하지 않은 문자를 명백하게 생성하는 다른 수단(기호 모드에서 단일 특정 문자와 모호하지 않게 연관된 키를 누르고 기호 모드에 진입하는 동작등과 같이)은, 특정 명확 문자(스페이스보다)는 삽입점(88)에서 출력 워드에 부가되는 것을 제외하고, 동일 결과를 얻는데 사용될 수 있다. 대안으로, 구두점 키(63)는 아래에 설명되는 바와 같은 마침표, 하이픈, 또는 어포스트로피를 생성하도록 눌러질 수 있다.

선택 리스트내의 제 1 엔트리가 키 스트로크 시퀀스의 소망의 해석이 아니다면, 사용자는 선택 키(60)를 반복적으로 누름으로써 선택 리스트내의 아이템을 통해 진행할 수 있다. 선택 키를 누르면, 점선으로 그려진 박스에 선택 리스트(76)의 제 1 엔트리를 다시 디스플레이하고, 또한 그 주위에 그려진 식별되는 포맷 박스로 삽입점(88)에서 제 1 엔트리를 다시 디스플레이한다. 개념상, 실선 박스에서 점선 박스로의 변경은 텍스트가 생성된 텍스트로 받아진 것으로, 선택 키를 사용자가 누름으로써 명백하게 선택된 것으로 가까워진다는 것을 나타낸다. 선택 리스트내의 제 1 엔트리가 키 스트로크 시퀀스의 소망의 해석이면, 사용자는 데이터 키(56)를 이용하여 다음 워드를 계속하여 입력한다. 선택 키가 스페이스 기능키로 오버로드되면, 스페이스는 다음 워드에 대한 텍스트를 삽입하기 전에 생성된다. 다른 방법으로, 다음 워드의 시작부는 보간 스페이스없이 현재 워드의 후부와 연결된다. 선택 키의 각각의 누름에 대하여, 선택 리스트내의 다음 엔트리는 점선으로 박스내에 들어가고, 엔트리의 카피는 삽입점에 임시 배치되고(이미 임시 배치된 워드를 대체), 점선으로 박스화된다. 다음 영역에 다음 엔트리를 임시 배치하면, 사용자는 선택 리스트를 반드시 참조하지 않고도 다음 영역에 주의할 수 있다. 사용자의 선택사양으로, 시스템은 또한 선택 키의 제 1 누름을 받자마자, 삽입점에 임시 배치된 워드는 현재 선택 리스트의 카피를 디스플레이하도록 확장(수직 또는 수평)할 수 있다. 사용자는 선택 리스트의 이러한 카피에 디스플레이되어야 하는 최대 워드 수를 선택할 수 있다. 대안으로, 사용자는 선택 키의 제 1 동작에 앞서, 삽입점에 항상 디스플레이되는 선택 리스트를 가진 것을 선택할 수 있다. 명확화 시스템은 현재 선택된 엔트리가 소망의 엔트리임의 확신하는 것으로서 다음 워드의 시작부(명백하게 모호하지 않은 문자의 생성 또는 모호한 데이터 키(56)의 동작에 의해 신호처리)를 해석한다. 선택된 워드는 사용자의 선택으로서 삽입점에 남아있고 경계 박스는 완전히 사라지고, 워드는 특정 포맷없이 규정 텍스트로 다시 디스플레이된다.

선택 리스트내의 제 2 엔트리가 소망의 워드이면, 사용자는 선택 키를 두번 누른 후 다음 워드를 입력하고, 명확화 시스템은 규정 텍스트로서 다음 영역에 제 2 엔트리를 배치한다. 제 2 엔트리가 소망의 워드가 아니면, 사용자는 선택 리스트를 검색하고, 다음 워드를 입력하기 전에 소망의 엔트리를 선택하기 위해 소망 횟수로 선택 키를 누른다. 선택 리스트의 끝단

에 도달할 때, 선택 키의 추가 누름 동작으로 선택 리스트는 스크롤되고 새로운 엔트리는 선택 리스트의 끝단에 부가된다. 선택 리스트의 상부에서의 이러한 엔트리는 사용자에게 디스플레이되는 리스트로부터 제거된다. 선택 키의 여러번의 누름 동작에 의해 선택된 엔트리는 사용자가 텍스트를 계속해서 누르기 위해 임의의 데이터 키(56)를 누를 때 텍스트 영역에 자동적으로 배치된다. 대안으로, 소망의 워드에 대응하는 키 스트로크 시퀀스의 엔트리를 수반하면, 사용자는 단순히 터치함으로써 선택 리스트로부터 소망의 워드를 선택할 수 있다. 선택 워드는 스페이스를 추가하지 않고 삽입점에 바로 출력되고, 선택 리스트는 제거된다. 사용자는 삽입점(88)에서 다음 영역으로 바로 출력되는 스페이스를 생성하기 위해 스페이스 키를 누를 수 있다.

대부분의 텍스트 엔트리에서, 키 스트로크 시퀀스는 워드를 형성하는 문자로서 사용자에 의해 의도된다. 그러나, 각각의 키와 관련된 복합 문자와 기호로 개별 키 스트로크와 키 스트로크 시퀀스는 몇몇 해석을 가질 수 있다는 것을 알 수 있다. 바람직한 감소된 키보드 명확화 시스템에서, 여러 상이한 해석은 자동적으로 결정되고 키 스트로크 시퀀스가 워드의 리스트로서 사용자에게 해석되고 디스플레이될 때 사용자에게 디스플레이된다.

예를 들어, 키 스트로크 시퀀스는 사용자가 입력할 수 있는 가능한 유효의 문자 시퀀스에 대응하는 워드 스템(이하 "스템 해석")의 용어로 해석된다. 워드 해석과 달리, 워드 스템은 불완전한 워드이다. 최종 키 스트로크의 가능한 해석을 지시함으로써, 사용자는 워드 스템에 의해 올바른 키 스트로크가 입력되었음을 쉽게 확인할 수 있고, 사람들의 태도가 워드의 중간에서 전환되었을 때 타이핑을 재개할 수 있다. 도 1에 도시된 바와 같이, 키 스트로크 시퀀스(ABC GHI DEF)는 유효 스템 "che"("check", "cheer"등의 워드에 앞선)과 "ahe"("ahead", "ahem"등의 워드에 앞선)를 형성하는 것으로서 해석된다. 그러므로, 스템 해석은 선택 리스트에서 엔트리(81,82)로서 제공된다. 바람직하게, 스템 해석은 데이터 키상의 추가 키 스트로크에 의해 각각의 스템으로부터 생성될 수 있는 모든 가능한 워드 세트의 복합 빈도수에 따라 분류된다. 디스플레이되어야 하는 이러한 엔트리의 최대 및 최소 복합 빈도수는 사용자에 의해 선택될 수 있거나, 본 시스템으로 구성될 수 있어서, 일부 스템 해석은 디스플레이되지 않을 수도 있다. 현재의 예에서, 스템 "bif" ("bifocals" 워드에 앞선)과, "bie" ("biennial"에 앞선)는 도시되어 있지 않다. 선택 리스트의 스템 해석을 리스트할 때, 스템 해석은 선택 리스트내에 도시되어 있는 워드를 중복하는 경우에 생략된다. 그러나, 그 스템이 생략될 때, 생략된 스템에 대응하는 워드는 이러한 워드를 그 스템으로서 가지는 긴 워드가 있음을 나타내기 위해 기호로 마킹될 수 있다. 스템 해석은 올바른 키 스트로크가 소망의 워드의 엔트리로 진행하도록 입력되었음을 확인함으로써 사용자에게 피드백을 제공한다.

감소된 키보드 명확화 시스템의 동작은 명확화 소프트웨어(108)에 의해 통제된다. 본 시스템의 바람직한 일실시예에서, 선택 키는 선택 리스트로부터 소망의 객체를 선택하는 기능을 모두 수행하고, 또한 적절할 때 명백한 스페이스 문자를 생성하도록 "오버로드"된다. 이러한 시스템에서, 플래그 "OverloadSelect"(블록 (164, 174a)를 참조)는 TRUE로 설정된다. 명백한 스페이스 문자를 생성하도록 정의된 개별 스페이스 키를 가진 시스템에서, 플래그 "OverloadSelect"는 FALSE로 설정된다.

도 3은 모호한 키스트로크 시퀀스를 명확하게 할 때 사용자를 도와주기 위해 선택 리스트를 생성하는 명확화 소프트웨어의 메인 루틴의 흐름도이다. 블록(150)에서, 본 시스템은 키보드(54)로부터 키 스트로크를 수신하는 것을 대기한다. 키스트로크를 수신하는 즉시, 블록(150A)에서, 본 시스템은 도 5A에 도시되고 아래에 상세히 설명되어 있는 구두점 키(63)를 필요로 하는 선처리동작을 수행한다. 결정 블록(151)에서, 수신된 키스트로크가 모드 선택 키인지를 결정하기 위해 테스트한다. 그러하다면, 블록(172)에서, 본 시스템은 현재 시스템 모드를 지시하는 플래그를 설정한다. 결정 블록(173)에서, 본 시스템 모드가 변경되었는지를 결정하기 위해 테스트한다. 그러하다면, 블록(171)에서, 키탑은 본 시스템 모드를 반영 할 필요가 있는 것으로서 재도안된다. 본 시스템 모드가 변경되지 않았다면, 또는 변경되어 키탑이 연속적으로 재도안되었다면, 남은 루틴은 블록(150)으로 진행하고, 다른 키 스트로크를 대기한다.

한편, 블록(151)에서 키스트로크가 모드 선택 키가 아니다면, 결정 블록(152)에서, 수신된 키스트로크가 선택 키인지를 결정하기 위해 테스트한다. 키 스트로크가 선택 키가 아니다면, 결정 블록(153)에서, 본 시스템은 명백한 숫자 모드와 같은 명백한 특수 문자 모드가 되는지를 결정하기 위해 테스트한다. 그러하다면, 결정 블록(166)에서, 임시적으로 선택된 아이템이 선택 리스트내에 존재하는지를 결정하도록 테스트가 행해진다. 그러하다면, 블록(167)에서, 아이템은 수용되고, 규정 텍스트로서 출력된다. 임시적으로 수용된 아이템이 선택 리스트내에 있지 않다면, 또는 존재하여 이미 수용되었다면, 블록(168)에서, 키 스트로크에 대응하는 명백한 문자는 텍스트 영역으로 출력된다. 다음, 결정 블록(169)에서, 기호 모드의 경우와 같이, 본 시스템 모드가 자동적으로 변경되었는지를 결정하기 위해 테스트된다. 그러하다면, 본 시스템 모드가 이전 동작 모드에 리턴되고 따라서 키탑이 재도안되는 블록(170, 171)로 진행한다. 그 다음 블록(150)에서 실행된다.

블록(153)에서, 명백한 문자 모드가 동작하지 않는다면, 블록(154)에서 키스트로크는 저장된 키 스트로크 시퀀스에 부가된다. 블록(156)에서, 그 키 스트로크 시퀀스에 대응하는 객체는 본 시스템의 어휘 모듈로부터 식별된다. 어휘 모듈은 키

스트로크 시퀀스와 연관된 객체 라이브러리이다. 하나의 객체가 검색된 키 스트로크 시퀀스를 근거로 하여 검색되는 임의의 저장 데이터 피스이다. 예를 들어, 어휘 모듈내의 객체가 숫자, 문자, 워드, 스템, 구문, 또는 시스템 기능 및 매크로를 포함할 수 있다. 이러한 객체 각각은 아래의 표에 간략하게 설명되어 있다.

객체	대응 데이터
숫자	단일 키스트로크에 대응하는 수, 각각의 디지트, 예를 들어, 2디지트 시퀀스 "42"
문자	한쌍의 키스트로크에 대응하는 문자 및 문자 시퀀스, 예를 들어, 3개의 문자 시퀀스 "str". 각 쌍의 키스트로크는 개별 문자를 입력하는 두개의 스트로크 명세 방법을 이용하여 명확하게 하는데 사용된다.
워드	단일 또는 복수의 키스트로크에 대응하는 워드, 예를 들어 4개의 문자 워드 "done"
스템	워드를 형성하는 긴 문자 시퀀스의 유효부를 나타내는 문자 시퀀스, 예를 들어, 워드 "albeit"의 스템으로서의 "albe"
구문	단일 또는 복수의 키스트로크에 대응하는 사용자 정의 또는 시스템 정의 구문, 예를 들어, "To Whom it May Concern"
시스템 매크로	시스템 또는 사용자 정의 기능을 나타내는 워드 및 관련 코드, 예를 들어, 현재 텍스트 영역을 제거하는 "clear". 표시한 워드에 추가로, 어휘 모듈에서 시스템 매크로 객체는 상기 기능을 수행하기에 필요한 실행가능 코드와 연관되어 있다.

바람직한 어휘 객체가 위에서 설명되었지만, 다른 객체가 논의될 수 있다는 것을 알 수 있다. 예를 들어, 그래픽 객체가 저장 그림 이미지와 관련될 수 있고, 또는 언어 객체가 저장된 언어 세그먼트와 연관될 수 있다. 일반적인 미스스펠링 워드와 타이핑 애러의 키 스트로크 시퀀스를 그 워드의 올바른 스펠링과 연결시키는 스펠링 객체로 또한 확대될 수 있다. 예를 들어, 문자 시퀀스 "ie" 또는 "ei"는 이러한 문자에 대한 키 스트로크가 적당한 시퀀스로부터 우연히 반대로 되더라도 워드 리스트에 나타날 수 있다. 처리를 간략하게 하기 위해, 각각의 어휘 모듈은 바람직하게 유사 객체를 포함하고 있다. 그러나, 여러 객체는 어휘 모듈과 혼합될 수 있다는 것을 알 수 있다.

단일 키(540)의 대표적인 다이아그램이 도 6에 도시되어 있다. 바람직한 실시예의 키의 내부 논리 표현은 물리적인 배열을 반영할 필요가 없다. 예를 들어, 541는 프랑스 어휘 모듈과 관련된 키의 바람직한 논리 표현이다. 이러한 추가 기호(542) (ÂÇÀ)는 프랑스 알파벳에서 필요하다. 또한, 그 기호는 프랑스 사전에서 감소하는 사용빈도수의 순서로 543에 바람직하게 색인되어 있다. 확대하면, 도 7은 논리 기호의 색인을 프랑스어의 키 누름의 명확화에 사용되는 키 색인과 관련시키는 바람직한 테이블이다.

워드 객체 어휘 모듈(110)의 대표적인 다이어그램은 도 10에 도시되어 있다. 트리 데이터 구조는 대응하는 키 스트로크 시퀀스를 근거로 하여 어휘 모듈내의 객체를 체계화하는데 사용된다. 도 10에 도시된 바와 같이, 그 어휘 모듈 트리내의 각각의 노드(N1, N2,...,N9)는 특정 키 스트로크 시퀀스를 표현한다. 트리내의 노드는 경로(P1,P2,...P9)에 의해 연결되어 있다. 명확화 시스템의 바람직한 실시예에 9개의 모호한 데이터 키가 있기 때문에, 그 어휘 모듈 트리내의 각각의 모 노드는 9개의 자 노드와 연결될 수 있다. 경로에 의해 연결된 노드는 유효 키 스트로크 시퀀스를 표현하고, 노드로부터의 하나의 경로 누락은 무효 키 스트로크 시퀀스, 즉, 임의의 저장 워드에 대응하지 않은 것을 나타낸다.

어휘 모듈 트리는 검색된 키 스트로크 시퀀스를 근거로 하여 검색된다. 예를 들어, 루트 노드(111)로부터의 제 1 데이터 키를 누르면, 제 1 데이터 키에 연관된 데이터가 루트 노드(111)의 안쪽으로부터 폐치되고, 평가되고, 노드(N1)로의 경로(P1)가 검색된다. 제 1 데이터 키를 누른 후 제 9 데이터 키를 누르면, 제 9 데이터 키에 관련된 데이터가 노드(N1)로부터 폐치되고, 평가되고, 노드(N19)로의 경로(P19)가 검색된다. 아래에 상세히 설명되는 바와 같이, 각각의 노드는 키 스트로크 시퀀스에 대응하는 다수의 객체와 관련되어 있다. 각각의 키 스트로크가 수신되고, 대응 노드가 처리되면, 하나의 객체 리스트는 키 스트로크 시퀀스에 대응하는 객체중에서 생성된다. 각각의 어휘 모듈로부터의 객체 리스트는 선택 리스트(76)를 생성하기 위해 본 명확화 시스템의 메인 루틴에 의해 사용된다.

도 8a는 각각의 노드와 관련된 바람직한 데이터 구조(400)의 블록도이다. 그 데이터 구조는 어휘 모듈 트리내의 자 노드에 각각의 모 노드를 연계시키는 정보를 포함하고 있다. 그 데이터 구조는 노드에 의해 표현되는 특정 키 스트로크 시퀀스와 관련된 객체를 식별하는 정보(구조)를 포함하고 있다.

노드 데이터 구조(400)의 제 1 필드는 노드에 의해 표현되는 특정 키 스트로크 시퀀스와 관련된 객체를 식별하는 정보와 9개의 가능한 키중 어느 것과 관련되어 있는지 그리고 모 노드에 연결된 자 노드의 일치성과 수를 표시하는 유효 키 비트 필드(402)이다. 바람직한 실시예에서 9개의 데이터 키가 있기 때문에, 최대 9개의 자 노드이 특정 모 노드에 연결되고, 9개의 유효 키 비트가 자 노드의 유무를 표시하는 유효 키 비트 필드에 제공된다. 각각의 유효 키 비트는 어휘 모듈내의 각각의 자 노드 데이터 구조에서의 포인터를 포함하는 포인터 필드(404a, 404b, ..., 404n)와 관련되어 있다. 자 노드와 관련된 키 스트로크가 모 노드와 관련된 키 스트로크 시퀀스의 유효 연속성이다면 자 노드만이 존재하기 때문에, 포인터 필드의 수는 각각의 노드에 대하여 가변한다. 예를 들어, 유효 키 비트 필드(402)는 가능한 9개의 키 스트로크 중 단지 6개만이 유효 자 노드를 유도한다는 것을 나타낼 수 있다. 단지 6개의 유효 경로가 있기 때문에, 6개의 포인터 필드만이 모 노드에 대한 데이터 구조에 포함된다. 유효 키 비트 필드(402)는 노드 데이터 구조내에 포함되어 있는 포인터 필드의 일치성을 확인하는데 사용된다. 키 스트로크가 유효 자 노드를 유도하지 않으면, 관련 포인터 필드는 어휘 모듈을 저장하는데 필요한 메모리 공간을 보전하기 위해 노드 데이터 구조로부터 생략된다.

노드에 의해 표현되는 키 스트로크 시퀀스에 대응하는 다수의 객체는 각각의 노드와 관련되어 있다. 객체 각각은 노드 데이터 구조에 포함되어 있는 유효 키 비트 필드(402)의 비트 패턴에 의해 표현되는 바와 같이 특정 유효 키에 부착된 패킷(408)내의 406의 명령에 의해 표현된다.

각 패킷(406)에서의 각 명령어는 각 노드에 의해 나타내어지는 키스트로크 시퀀스에 대응하는 객체중의 하나를 설명한다. 객체의 설명은 2개의 객체 리스트의 유지를 요구한다. 도 11은 부모와 자식으로부터의 명확화 소프트웨어 프로세스에 의해 어휘 모듈 트리에서 다이내믹하게 만들어진다. 객체 리스트(430)는 2개의 키스트로크를 나타내는 노드와 관련된 객체($1-N_1$)를 포함하는 객체 리스트이다. 객체 리스트(440)는 3개의 키스트로크를 나타내는 노드와 관련된 객체($1-N_2$)를 포함하는 객체 리스트이다. 각 객체 리스트는 각 노드와 관련된 모든 객체의 리스트를 포함한다. 객체 리스트(430)는 도 1의 키보드로부터의 키스트로크 스퀀스(ABC ABC)를 나타내는 부모 노드와 관련된다. 객체 리스트(440)는 키스트로크 스퀀스(ABC ABC TUV)를 나타내는 자식 노드와 관련된다. 객체 리스트의 크기는 각 노드와 관련된 객체의 실제수를 계산하기 위해 바뀐다.

자식 노드와 관련된 각 객체는 문자 시퀀스를 부모노드에 대해 구성된 객체상에 추가함으로써 구성된다. 따라서, 도 8A의 명령어 패킷(406)은 자식 노드 객체를 구성하기 위해 사용된 객체를 부모 노드 객체 리스트로부터 식별하는, 도 8B에 도시된 OBJECT-LIST-INDEX 필드(556)를 갖는 명령어(558)를 포함한다. 예를 들면, 도 11을 참조로, 구 객체 리스트(430)에 있는 제 1 객체("ba")는 새로운 객체 리스트(440)에 있는 제 2 객체("bat")를 구성하기 위해 사용된다. 따라서, 이전의 객체 식별자 필드(OBJECT-LIST-INDEX;556)는 링크를, 새로운 객체를 구성하기 위해 사용된 구 객체를 식별하기 위해, 구 객체 리스트에 있는 엔트리에 제공한다.

명령어(558)는 새 객체를 구성하기 위해, 식별된 객체에 추가하는 기호를 나타내도록 LOGICAL-SYMBOL-INDEX 필드(555)를 또한 포함한다. 따라서, LOGICAL-SYMBOL-INDEX 필드는 새로운 객체를 구성하기 위해 추가될 노드의 키 시퀀스에 있는 최종 키로부터의 문자를 특정한다. 문자는 도 7에 도시된 바와 같은 테이블에 의해 특정되며, 여기에서, LOGICAL-SYMBOL-INDEX 필드는 테이블(550)의 제 1 열에 있는 논리 기호 인덱스(552)를 대응시키며, 특정 키가 나타나는 열은 테이블의 제 1 행에 있는 주어진 키 인덱스에 의해 식별된다. 예를 들면, 도 11을 참조로, 새로운 객체 리스트(440)에 있는 제 1 객체("CAT")는 구 객체 리스트(430)에 있는 제 2 객체("CA")를 사용함으로써, 또한 "T"를 특정하도록 추가 키스트로크를 부가함으로써 구성된다. 따라서, 도 7의 논리 기호 인덱스 테이블에서, "T"는 TUV 키상에 있는 제 1 논리 문자이며, 객체"CAT"를 생성한 명령어의 LOGICAL-SYMBOL-INDEX 필드는 테이블에 있는 제 1 문자를 지시하기 위해 1로 세트된다. 이러한 방식으로 개체를 부호화하는 것은 각 노드와 관련된 종래의 키 시퀀스, 및 키와 문자의 종래의 관련성의 사용이, 각 어휘 모듈에 요구되는 저장스페이스의 양을 많이 축소하도록 한다.

또한 어휘 암호 기술은 검색없이 어휘 모듈 엔트리에 접근하게 한다. 각각의 새로운 유효 키스트로크의 개념에 있어, 시스템은 구 객체 리스트로부터 새로운 객체 리스트를 구성하기 위해, 현재 노드에서 키와 관련된 명령어를 실행한 후, 단일 포인터를 적절한 자식 노드로 뒤따르게 한다. 또한, 새로운 객체는 구 해석상에 추가하기 위해, 어휘 모듈에 있는 모든 객체를 저장하지 않고, LOGICAL-SYMBOL-INDEX 필드를 사용하여 정의된다. 따라서, 어휘 모듈에 있는 복수의 객체에 의해 공유된 어간은 단지 한번 저장되며, 어간으로부터 도출된 모든 객체를 만들기 위해 사용된다. 개시된 저장방법은 자식 노드의 객체 리스트를 구성하기 위해, 어휘 모듈 트리에 있는 부모노드로부터 객체 리스트를 유지하는 것을 요구한다.

도 7에서 설명된 바와 같이, 논리 기호 인덱스 테이블에 있는 엔트리는 단일 문자일 것을 요구하지 않으며, 임의의 시퀀스는 단일 엔트리를 점유할 수 있다. 예를 들면, ASCII 문자열 "tlike"는 "catlike"를 형성하기 위해, 구 객체 리스트로부터 제 2

객체 "ca"에 추가될 수 있다. 이와 같은 방식으로, 입력된 키스트로크 시퀀스의 길이는 관련된 객체의 길이에 필수적으로 직접 일치하지 않는다. 기호 인덱스 테이블의 엔트리에 저장된 ASCII 시퀀스는 어휘 객체가 임의의 키 시퀀스, 즉, 어휘 모듈 트리내에 있는 임의의 위치에 저장된, 임의의 키 시퀀스에 의해 식별되도록 하게 한다.

임의의 키스트로크 시퀀스를 갖는 객체를 저장하는 능력은 생략 및 축소의 시스템 프로세싱의 속력을 높이기 위해 사용된다. 생략 및 축소는 구두를 무시하면서, 이들의 순수 영문자 내용에 대응하는 키스트로크 시퀀스에 의해 식별될 수 있다. 이 결과는 생략 및 축소가 구두의 입력 없이도 사용자에 의해 쉽게 액세스되게 하는 것이며, 키스트로크에 상당한 보류를 낳는다. 예를 들면, 사용자는 영문자를 "n" 및 "t" 사이에 타이핑함이 없이 "didn't"에 대한 키스트로크 시퀀스를 입력할 수 있다. 키스트로크 시퀀스 "didn't"에 대응하는 어휘 모듈에 있는 명령어는 "n" 및 "t" 사이에 있는 어포스트로피를 갖는 ASCII 시퀀스를 테이블내에 있는 단일 기호로서 참조한다. 따라서, 명확화 시스템은 사용자가 구두점을 입력 할 필요없이 올바른 단어 "didn't"를 사용자에게 자동적으로 디스플레이 한다. 명확화 시스템은 고유문자를 갖는 외국어(TUV 키상에 있는 단일 키스트로크로서 입력될 수 있는 "?quot; 와 같은 것)를 적절히 디스플레이 하기위해 동일한 테이블을 사용한다. 대문자는 동일한 방식으로 처리될 수 있다. 처음 대문자, 또는 중간부분에서의 대문자를 갖는 모든 대문자에서 항상 사용될 수 있는 워드는 사용자가 이러한 대문자를 입력할 필요를 없애면서, 대문자를 지시하는 키스트로크를 빠트리는 키스트로크 시퀀스와 관련될 수 있다. 또한 객체 타입 필드는 구성된 객체에 대한 추가 정보를 특정하기 위해, 각 명령어(558)내에 포함될 수 있다. 객체 타입 필드는 생성된 객체가 워드, 어간, 또는 임의의 다른 객체인지를 특정하기 위해 코드를 포함할 수 있다. 따라서, 객체 타입 필드는 다른 타입의 객체가, 주어진 어휘 모듈내에서 혼합되도록 하게 한다. 더우기, 객체 타입 필드는 워드의 부분속도에 관한 정보, 객체가 어떻게 대문자화되었는지에 대한 정보, 또는 다양한 어형변화 및 어미를 구성하기 위해 필요한 정보를 또한 포함할 수 있다. 부분속도정보를 갖는 어휘 모듈을 사용하는 축소된 키보드 명확화 시스템은 명확화 프로세스를 향상하도록 신팩스 분석을 구현하기 위해 추가의 정보를 사용할 수 있다. 객체 타입 필드는 텍스트를 압축된 형태로 전송하기 위해 단일 코드를 또한 포함할 수 있다. 단일 코드는 입력된 키스트로크 시퀀스 또는 관련된 명확화 문자를 전송하는 대신에, 원격 터미널에 전송될 수 있다.

바람직한 어휘 모듈 트리 데이터 구조의 키 특성중의 하나는 각 노드와 관련된 객체가 그들의 사용회수에 따라서, 노드 데이터 구조(400)에 저장된다는 것이다. 즉, 패킷(406)에 있는 제 1 명령어에 의해 구성된 객체는 406에 있는 제 2 명령어(존재한다면)에 의해 구성된 객체보다 더 높은 사용회수를 가지며, 제 3 명령어(존재한다면)에 의해 구성된 객체보다 더 높은 사용회수를 갖는다. 이러한 방식으로, 객체는 자동적으로 객체 리스트내에 위치되어, 그 결과, 이들은 사용회수의 증가에 따라서 분류된다. 이러한 설명의 목적에 대해, 워드 객체의 사용회수는 사용의 대표적인 전체내에 있는 주어진 워드의 사용 가능성에 관련되며, 이것은 각 워드가 전체내에서 발생한 것의 복수 배에 비례한다. 어간 객체의 경우에서, 사용회수는 동일한 어간을 공유하는 모든 워드의 회수를 합함으로써 결정된다. 사용회수 또는 각 노드에서의 다른 랭크 정보를 저장하는 것은 시스템이 사용중일때, 각 객체의 랭크를 분류하며 결정하는 필요성을 피하게 한다. 이것은, 저장된 객체가 매우 큰 수의 더 긴 워드에 공통인 공유어간을 포함할 수 있기 때문에, 워드 객체 어휘에서 중요한 내포를 갖는다. 이들 어간의 상대적인 랭크를 결정하는 것은 수송 가능한 컴퓨터 디바이스에 대한 상당한 프로세싱 오버헤드를 추가하면서, 자식노드의 전체 트리를 주사하는 것을 다이나믹하게 요구하며, 각 어간에 대한 정보를 축적한다. 미리 정보를 결정하며, 정보를 어휘 데이터에 저장하는 것은 프로세싱 오버헤드를 줄인다. 더우기, 사용 회수 또는 랭크는 노드에 있는 객체(406)의 순서에 의해 내포적으로 표현될 때, 이러한 정보에 대해 추가의 저장 스페이스가 필요없게 된다.

바람직하게 객체는 사용회수에 따라서 노드 데이터 구조(400)내에 순서적으로 저장되는 반면, 사용 필드의 회수는 각 명령어와 관련될 수 있다. 사용 필드의 회수는 관련된 객체의 사용회수와 대응한 대표 수를 포함한다. 다른 객체간의 사용회수는 각 객체의 사용필드의 회수를 비교함으로써 결정된다. 각 객체 패킷을 갖는 사용필드의 회수와 관련한 후자의 구성을 사용하는 이점은 명확화 시스템에 의해, 사용 필드의 회수를 변화시킬 수 있다는 것이다. 예를 들면, 시스템은 사용자가 대표적인 텍스트 엔트리 동안에 어휘 모듈에 있는 일부 객체를 사용한 회수를 반영하기 위해, 사용필드의 회수를 변화시킨다.

도 3으로 돌아가서, 블록(156)에서, 수신된 키스트로크 시퀀스에 대응하는 이들 객체는 각 어휘 모듈에서 식별된다. 도 12는 대응 객체를 특정 어휘 모듈에서 식별하기 위해 수신된 키스트로크 시퀀스를 분석하기 위한 서브루틴(600)의 플로차트이다. 서브루틴(600)은 특정 키스트로크 시퀀스에 대한 객체 리스트를 구성한다. 블록(602)은 새로운 객체 리스트를 제거한다. 블록(604)은 트리(110)의 운행을 루트 노드(111)에서 개시한다. 블록(604)은 제 1 키 프레스를 얻는다. 블록(608)은 내지 612)은 모든 이용가능한 키 프레스를 프로세스하기 위해 루프를 형성한다. 블록(608)은 도 13에 있는 서브루틴(620)을 호출한다. 결정 블록(610)은 모든 이용가능한 키 프레스가 프로세스되었는지를 결정한다. 만약 임의의 키 프레스가 프로세스되지 않은 상태에 있다면, 블록(612)은 다음 이용가능한 키 프레스로 나아간다. 만약 모든 키 프레스가 프로세스되었다면, 블록(614)은 완성된 객체 리스트로 복귀된다. 만약 주 루틴이, 새로운 키스트로크 시퀀스를 사용하여 반복적으로

서브루틴(600)을 호출한다면, 최종 키, 및 최종키를 제외한 모든 키보다 하나 더 큰 키를 갖는 각각은 이전 호출에서와 동일하며, 그후, 만약, 서브루틴(620)이, 단지 가장 최근의 키 프레스를 프로세스하기 위해 직접 호출된다면, 초기화 블록(602,604)은 바이패스될 것이다.

도 13은 서브루틴(600)으로부터 호출된 서브루틴(620)의 플로차트이다. 도 3에 도시된 주 루틴에서, 키스트로크는 시스템에 의해 블록(150)에서 검지된다. 만약 유효 패스가, 키스트로크에 대응하는 자식에 존재한다면, 새로운 키스트로크의 수신은 어휘 모듈 트리에 아래방향으로 운행을 하게 한다. 따라서, 도 13에 있는 블록(621)에서, 노드(400) 데이터 구조의 유효 키 비트 필드는 유효 명령어 및 지시기가 수신된 키스트로크에 대응하는지를 결정하기 위해 실행된다. 결정 블록(621)에서, 테스트는 404a와 같은 지시기 필드 및 명령어(406)로 이루어진 유효 패킷(408)이 입력된 키스트로크에 대응하여 존재하는지를 결정하기 위해 유효 키 비트로 만들어진다. 만약, 키스트로크에 대응하는 유효 패킷이 없다면, 수신된 키스트로크는 어휘 모듈내에 있는 임의의 객체에 대응하지 않는 무효 키스트로크 시퀀스의 부분이기 때문에, 블록(624)에서, 구 객체 리스트는 선택 리스트를 생성하기 위해 주 루틴에 복귀한다. 따라서, 블록(622,624)을 포함하는 서브루틴(620)의 브랜치는 임의의 무효 키스트로크 시퀀스를 무시하며, 명확화 시스템에 의해 생성된 선택 리스트에서의 가능한 포함에 대해 부모 노드에서 생성된 객체 리스트를 복귀시킨다.

유효 패킷이 결정 블록(622)에서, 수신된 키스트로크에 대응하여 존재한다면, 서브루틴은 새로운 객체 리스트가 구 객체 리스트로 복사되는 블록(626)으로 나아간다. 위에서 설명한 바와 같이, 새 객체를 구성하기 위해, 명확화 시스템은 구 객체 리스트의 복사를 개시한다. 따라서, 블록(626)에서, 이전에 노드로부터의 객체 리스트는 새로운 객체 리스트를 구성하기 위해 사용될 수 있도록 저장된다.

블록(628)은 주어진 키와 관련된 제 1 유효 명령어를 폐치한다. 블록(630)은 반복자(NEW-INDEX)를 1로 개시하며, 그 결과, 제 1 명령어는 새로운 객체 리스트에 있는 제 1 아이템을 생성할 것이다. 다음, 서브루틴은 유효 명령어와 관련된 객체 리스트를 구성하기 위해서, 블록(632 내지 642)로 포함된 루프를 입력한다. 블록(632)에서, OBJECT-LIST-INDEX 필드(556)는 실행되며, 대응 객체는 구 객체 리스트로부터 로딩된다. 블록(634)에서, LOGICAL-SYMBOL-INDEX 필드(555)는 실행되며, (도 7에 도시된 550과 같은 논리 기호 인덱스 테이블을 통하여 수신된 키스트로크와 관련된)적절한 기호는 식별된 객체의 종료에 추가된다. 만약, 주어진 키(551) 및 논리 기호 인덱스(552)에서의 기호 테이블(550)에 있는 엔트리가 문자 시퀀스를 유지한다면, 1 보다 큰 길이의 ASCII 시퀀스는 블록(634)에 식별된 객체에 추가될 수 있다. 블록(636)에서, 결합된 객체 및 기호는 새로운 객체 리스트에 있는 새로운 객체로서 저장된다. 블록(638)에서, 테스트는 서브루틴이, 주어진 노드에서의 주어진 키와 관련된 최종 유효 명령어를 프로세스하는지를 결정하기 위해 만들어진다. 만약 최종 유효 명령어가 프로세스하지 않았다면, 블록(640)에서, 다음 유효 명령어가 폐치된다. 블록(642)에서, NEW-INDEX는 증분된다.

만약 결정 블록(638)에서의 테스트가 노드에 대해 모든 객체가 구성되도록 지시한다면, 서브루틴은 블록(644)으로 진행하게 하며, 관련된 지시기를 자식 노드로 뒤따르게 한다. 블록(646)에서, 새로운 객체 리스트는 선택 리스트를 생성하기 위해서 주 루틴으로 복귀된다. 각 노드와 관련된 객체 리스트를 생성하는 서브루틴(600)은 사용자로부터 수신된 각 키스트로크에 대하여 수행된다. 각 키스트로크는 어휘 모듈 트리내에 있는 서브루틴 1 추가 레벨로 단순히 나아가기 때문에, 어휘 모듈의 "검색"은 사용자가 새로운 키스트로크 시퀀스를 입력할 때는 수행되지 않는다. 검색은 각 키스트로크에 대해서는 수행되지 않기 때문에, 어휘 모듈은 최소한의 프로세싱 오버헤드를 가진 각 노드와 관련된 객체의 리스트를 복귀시킨다.

어휘 모듈 객체 및 키스트로크 시퀀스사이의 관계는 어휘 모듈의 상세한 구현이다. 현재의 입력 키 시퀀스와 관련된 노드를 프로세싱할 때, 추가 자식 노드는 입력된 키스트로크 시퀀스에서 시작한 키스트로크 시퀀스를 갖는 후보객체를 식별하기 위해 주사될 수 있으며, 그것은 어떤 임계값보다 상대적으로 더 큰 회수로 발생한다. 이러한 임계값은 이것이 선택리스트 영역(70)을 디스플레이상에 채우기에 충분한 객체를 생성하는지 아닌지와 같은 현재 노드의 특성을 근거로 하여 다이나믹하게 조절될 수 있다. 객체가 식별될 때까지, 유효 경로를 따라 어휘 모듈 트리의 아래 방향으로 주사함으로써 식별될 수 있다. 이러한 후보 객체는 한번에 식별될 수 있으며, 데이터베이스는 입력 워드 리스트로부터 구성되며, 후보 워드의 완성에 대응하는 노드 및 명령어는 노드의 운행이 후보 워드의 생성에 대응할 때를 시스템이 인식하도록 하는 방법으로 마킹될 수 있다. 동일한 노드 및 명령어는 트리 구조가 아래 설명된 압축 프로세스에서 변형된 후, 다른 워드에 대응하는 다른 입력 키 시퀀스 처리에서 주사될 수 있으므로 특정 마킹이 요구될 수 있다. 오직 충분한 정보만이, 키 시퀀스에 있는 종래 위치에서의 특정 키 값 또는 시퀀스 길이와 같이, 다른 시퀀스와 후보 워드 시퀀스를 구별하기 위해 사용될 수 있다. 대안적으로, 후보 워드에 대한 터미널 노드는 이들이 아래 설명된 바와 같이, 다른 워드에 대해 사용된 노드와 합병하지 않는 방식으로 특정하게 마킹될 수 있다. 이러한 명확화 소프트웨어는 이들 후보 객체에 대한 현재 노드의 파생의 한정된 이웃 내에서 앞을 향하여 검색될 수 있으며, 그후, 검색된다면, 객체에 대응하는 모든 키스트로크가 입력되기 전에, 선택리스트

에 위치될 수 있다. 이들 객체는 입력 키스트로크 시퀀스와 직접 관련된 객체와 함께 포함된다. 선택 리스트("예견능력" 특징으로 일컫는)에 있는 더 긴 키스트로크 시퀀스와 관련된 객체를 디스플레이하는 것은 객체를 특정하도록 유지 키스트로크를 완료함이 없이, 사용자가 즉시 객체를 선택적으로 선택할 수 있게 한다.

도 3을 참조로, 블록(158-165)에서, 어휘 모듈에 있는 키스트로크 시퀀스를 조사함으로써 탐색된 객체는 우선순위로 되며, 선택 리스트(76)에 있는 사용자에게 디스플레이된다. 선택 리스트에서 디스플레이된 객체의 시퀀스를 결정하기 위해, 우선순위는 각 어휘 모듈사이에서, 및 각 어휘 모듈로부터의 복귀된 객체 사이에서 확립된다. 다양한 어휘 모듈로부터 식별된 객체 리스트를 우선순위로 하기 위해, 블록(158)에서, 축소된 키보드 명확화 시스템의 동작 모드가 실행된다. 위에서 설명된 바와 같이, 동작의 정상 모드에서, 해석은 선택 리스트에 먼저 디스플레이된다. 따라서, 워드 어휘 모듈로부터의 객체 리스트는 다른 어휘 모듈로부터의 객체 리스트보다 더 높은 우선순위를 부여받는다. 역으로, 만약, 명확화 시스템이 동작의 수치적인 모드에 있다면, 수치 해석은 다른 어휘 모듈보다 더 높은 순위를 부여받을 것이다. 따라서, 명확화 시스템의 모드는 어휘 모듈 객체 리스트 사이의 우선순위를 규정한다. 일정 모드에서, 어휘 모듈로부터의 객체 리스트는 선택 리스트 전체로부터 빠질 수 있다.

어휘 모듈로부터 생성된 객체 리스트는 단일 엔트리만을 포함할 수 있거나, 다중의 엔트리를 포함할 수 있다. 따라서, 블록(160)에서, 동일한 어휘 모듈로부터의 객체 사이의 우선순위는 객체 리스트가 다중의 엔트리를 포함한다면 해결된다. 주어진 어휘 모듈에서 조사된 특정 키스트로크 시퀀스를 매칭하는 객체는 서로에 대해 이들의 상대적인 프레젠테이션을 결정하는 우선순위가 또한 주어진다. 위에서 설명된 바와 같이, 디폴트 프레젠테이션 순서는 사용회수를 대표적인 전체 사용에서 감소시킴에 의한 것이 바람직하다. 각 객체와 관련된 우선순위 데이터는 선택 리스트에 있는 객체를 순서화하기 위해 사용된다. 선택 리스트 영역(70)은 디스플레이될 수 있는 엔트리의 수에 제한되기 때문에, 소정의 최소 사용회수 아래로 떨어진 객체는 선택 리스트의 초기 디스플레이로부터 빠진다. 사용자가 디스플레이될 리스트의 종료를 초과하여 스크롤할 때, 빠진 객체는 뒤에, 선택리스트에 추가될 수 있다. 선택 리스트는 자동으로 스크롤되어, 현재 선택된 객체는 항상 볼 수 있다. 또한 사용자는 추가 객체를 뷰내에 수동으로 스크롤하기 위해서 전용 스크롤 버튼을 사용할 수 있으며, 이경우 현재 선택된 객체는 뷰 밖으로 스크롤할 수 있다. 대안적으로 선택 리스트에 있는 모든 객체는 사용자 요구로 "드롭-다운"리스트에 동시에 디스플레이될 수 있다.

어휘 모듈에서 조사된 객체의 프레젠테이션과 관련된 복수의 특성은 적절한 시스템 메뉴를 액세스함으로써 사용자-프로그램 가능하다는 것이다. 예를 들면, 사용자는 선택 리스트 영역에 있는 객체의 클래스 또는 개별 객체의 순서를 특정할 수 있다. 또한, 사용자는 각 어휘 모듈로부터 식별된 객체사이, 및 어휘 모듈 사이의 우선 순위를 결정하는 우선순위 레벨을 세트할 수 있다. 이러한 방식으로, 선택 리스트 영역에서 사용자에게 공급되는 엔트리의 수는 최소로 유지될 수 있다. 선택 리스트 영역에서의 추가 엔트리는 선택 키의 반복 누름에 의해 뷰로 항상 스크롤될 수 있다.

객체사이의 우선순위가 해결된 후에, 블록(165)에서, 선택 리스트는 식별된 객체로부터 구성되며, 사용자에게 공급된다. 모호한 키스트로크 시퀀스의 디폴트 해석이 사용자에 의해서 입력될 때, 선택 리스트에 있는 제 1 엔트리는 일시적으로 포스트되며, 테스트 영역(66)에 있는 삽입 포인트(88)에서 강조된다. 명확화 소프트웨어 루틴은 그후 다음 키스트로크를 기다리기 위해, 블록(150)으로 복귀한다.

결정 블록(152)에의 복귀함과 함께, 만약 검지된 키스트로크가 선택 키(60)라면, "yes"브랜치가 결정 블록(152)으로부터 결정 블록(163)에 잡히며, 만약, 현재 선택 리스트가 비어 있다면, 테스트는 결정된다. 그래서 만약, 블록(164)에서, OverloadSelect 플래그가 TRUE로 세트되면, 명시 스페이스가 생성되며, 텍스트 영역으로 즉시 출력된다. OverloadSelect 플래그는 명시 스페이스 키(64)를 포함하지 않는 시스템에서 TRUE로 세트되는 시스템 플래그이며, 임의의 연속 시퀀스의 선택 키 구동의 제 1 구동에 대해 생성되거나, 또는 만약, 선택 리스트가 비어 있다면, 선택 키가 즉시 구동된다. 수신된 키스트로크의 이러한 초기 프로세싱 후, 블록(164B)에서, 시스템은 도 5B에 도시된 구두 키을 위하여 필요한 포스트 프로세싱을 수행하며, 아래 설명되어 있다. 다음, 실행은 블록(150)으로 복귀한다. 만약, 결정 블록(163)에서, 선택이 비어 있지 않으면, "no"브랜치가 블록(174A)에 잡힌다. 블록(174A)에서, 만약, OverloadSelect 플래그가 TRUE로 세트되면, 스페이스는 선택 리스트에 및 삽입 포인트에 있는 각 텍스트 아이템의 종료에 추가된다. 블록(174)에서, 선택 리스트(또한, 일시적으로 포스트된 삽입 포인트에서)에 있는 제 1 엔트리주변의 입체-라인 박스는 점선 박스로 변한다. 도 3에 도시된 블록(175)에서, 그후 시스템은 사용자에 의해 입력된 다음 키스트로크를 검지하기 위해 기다린다. 키스트로크를 수신하자마자, 그 후, 블록(175A)에서, 시스템은 도 5A에 도시된 구두 키(63)에 요구되는 프리-프로세싱을 수행하며, 아래 설명되어 있다. 결정 블록(176)에서, 테스트는 다음 키가 선택키인지를 결정하기 위해 만들어진다. 만약 다음 키가 선택키라면, 블록(178)에서, 시스템은 선택 리스트에 있는 다음 엔트리로 나아가며, 시스템을 현재 선택된 아이템으로서 마킹한다. 수신된 키스트로크의 초기 프로세스에 이어서, 블록(178B)에서, 시스템은 도 5B에 도시된 구두 키(63)에 요구되는 프리-프로세싱을 수행하며, 아래 설명되어 있다. 블록(179)에서, 현재 선택된 엔트리는 일시적으로 선택 리스트 및 엔

트리 주변의 점선 박스를 갖는 삽입 포인트에 디스플레이된다. 다음, 루틴은 사용자에 의해 입력된 다음 키스트로크를 검지하기 위해서, 블록(175)으로 복귀한다. 블록(175-170)에 의해 형성된 루프는 사용자가, 선택 키를 여러번 내리누름으로 써 더 적은 사용회수를 갖는 입력된 모호한 키스트로크 시퀀스의 다양한 해석을 선택할 수 있게 한다.

만약 다음 키스트로크가 선택키가 아니라면, 루틴은 결정 블록(176)으로부터, 일시적으로 디스플레이된 엔트리가 키스트로크 시퀀스 해석으로서 선택되며, 텍스트 영역에서 포맷하는 정상 엔트리로 변환하는 블록(180)으로 계속된다. 블록(184)에서, 선택 키 다음에 오는 모호한 키스트로크의 수신은 새로운 모호한 시퀀스의 시작을 시스템에 지시하기 때문에, 구 키스트로크 시퀀스는 시스템 메모리에서 제거된다. 새롭게 수신된 키스트로크는 후에, 새로운 키스트로크 시퀀스를 시작하기 위해 블록(154)에서 사용된다. 더 높은 사용회수를 갖는 워드 해석은 디폴트 선택으로 제공되기 때문에, 명확화 소프트웨어의 주 루틴은 사용자가, 선택 키의 추가 구동이 필요한 최소 수의 인스턴스를 갖는 텍스트를 연속적으로 입력할 수 있게 한다.

위에서 언급한 바와 같이, 구동의 정상 모드에서, 워드에 대응하는 선택 리스트(76)에 있는 엔트리는 리스트에 먼저 제공된다. 다른 순환에서, 리스트에 먼저 공급되는 다른 키스트로크 시퀀스 해석을 갖는 것이 바람직할 수 있다. 예를 들면, 리터럴 선택 "리스트"의 디스플레이를 디스플레이 스크린상에 제공하기에는 충분하지 않은 디스플레이 영역이 있는 시스템에서, 데이터베이스로부터의 매칭가능한 객체의 세트는 소프트웨어에 의해 내부적으로 저장되며, 디폴트 객체 또는 명시 선택 객체(선택 키의 하나 이상의 구동을 따라가게 하는)만이 삽입 포인트에 디스플레이된다.

상기 시스템에서, 새로운 키스트로크 시퀀스에 대응하는 가장 빈번한 워드는 이전 입력시퀀스에 대응하는 가장 빈번한 워드와 거의 또는 전혀 닮지 않았기 때문에, 삽입 포인트에서 디스플레이된 객체는 입력 시퀀스에 추가된 각 키스트로크와 함께 상당히 변하는 경향이 있다. 그 결과는 삽입 포인트에 디스플레이된 객체가 사용자에게 "급변함"을 나타내게 하며, 일정하게 변하는 양상은 친숙하지 않은 사용자 또는 새로운 사용자에게 특히 혼란스럽게 될 수 있다. 삽입 포인트에서의 이러한 "급변성"을 줄이기 위해, 시스템은 현재의 키 시퀀스에 대응하는 가장 빈번한 워드 객체를 항상 디스플레이하는 것보다는 차라리, 만약 객체가 임계값을 초과하는 전체 회수를 갖는 더 긴 워드 또는 복수의 워드에 대응한다면, 워드, 또는 어간 객체가 디스플레이될 수 있게 구성될 수 있다. 이러한 "어간촉진"의 결과는 삽입포인트에서 디스플레이된 워드에서의 변화가 추가문자를 이전에 디스플레이된 텍스트 객체에 단순한 추가로 빈번히 이루어진다는 것이다. 예를 들면, 빈번한 워드 "this"를 타이핑함에 있어서, 사용자는 4개의 키스트로크(TUV,GHI,GHI,PQRS)를 누른다. 어간 촉진을 수행함이 없이, 시스템은 객체의 연이은 시퀀스(한번에 한번)를 삽입 포인트:"t" "vi" "ugh" "this"에 디스플레이한다. 어간 촉진이 수행될 때, 시스템은 "t" "vi" "ugh" "this"를 디스플레이한다. 그 결과는 새로운 또는 친숙하지 않은 사용자가, 명확화 시스템이 그 또는 그녀의 키스트로크를 올바르게 해석한다는 더 많은 신뢰성을 갖게 한다는 것이다. 도 4의 플로차트는 주어진 키스트로크 시퀀스에 대해 주어진 노드에서 디스플레이하기 위해 객체를 시스템이 어떻게 결정하는지를 보여주는 것이다.

도 4는 각 노드가 모호한 키스트로크 시퀀스에 대응하는 노드의 트리구조에 있는 워드의 데이터베이스를 생성하는 소프트웨어 프로그램 루틴의 플로차트이다. 각 노드에서, 복수의 워드 객체는 키스트로크 시퀀스에 의해 생성된 가능 워드 또는 어간에 대응하여 생성된다. 블록(190)에서, 임계값"MinFactor"이 세트된다. 이 임계값은 텍스트 객체의 전체 회수가 워드 객체이상을 초과하도록 촉진하기 위해(즉, 노드에 대한 선택 리스트에서 먼저 디스플레이되는 디폴트 객체가 되기 위해) 노드에서의 가장 빈번한 워드 객체의 전체회수를 초과해야 한다. 워드 객체의 전체 회수는 워드 자체의 회수에, 워드가 어간인 더 긴 모든 워드(즉, 더 긴 워드의 초기 문자에 대응하는) 회수의 합을 더한 것이다. 블록(191)에서, 제 2 임계값 "MaxWordFreq"은 다른 어간이 촉진될 수 있는 것 이상의 워드의 상대적인 최대 회수에 대응하는 것을 세팅한다. 블록(192)에서, 트리에 있는 모든 노드는 주사되며, 프로세스되지 않은 것으로 마킹된다. 블록(193)에서, 루틴은 트리의 모든 노드가 프로세스된 것인지를 결정한다. 만약 그렇다면, 루틴은 정지하고, 그렇지 않다면, 블록(194)에서, 시스템은 다음 프로세스되지 않은 노드를 폐치하며, 최대로 빈번한 워드 객체의 회수(MaxWordFreq), 어드가 어간인 모든 워드를 함께 워드의 전체 회수(MostFreqWordTot), 및 객체가 어간인 모든 워드의 전체회수(MostFreqStemTot)가 노드와 관련된 모든 객체중의 가장 높은 객체를 결정한다. 블록(195)에서, 시스템은 MostFreqStemTot가 MinFactor의 적어도 하나의 인자에 의해 MostFreqWordTot값을 초과하는지를 테스트한다. 만약 그렇지 않다면, 노드와 관련된 디폴트 객체는 변하지 않은 상태에 놓이며, 시스템은 노드가, 블록(193)으로 복귀되기 전에, 이미 프로세스된 것으로서 마킹된다. 만약 블록(195)에서, MostFreqWordTot이 MinFactor의 적어도 하나의 인자에 의해 초과된다면, 그 후 블록(196)에서, 시스템은 노드와 관련된(MostFreqWord) 현재 디폴트 객체가 다른 객체가 촉진될 수 있는 이상의 워드의 최대값(MaxWordFreq)을 초과했는지를 테스트한다. 만약 그렇다면, 노드와 관련된 디폴트 객체는 변하지 않고 유지되며, 시스템은 블록(198)으로 복귀된다. 만약 그렇지 않다면, 블록(197)에서, 노드는 변형되어, MostFreqStemTot와 관련된 객체는 블록(198)으로 복귀되기 전, 노드와 관련된 디폴트 객체로서 지정된다.

매우 일반적인 어간이 공통 워드에 걸쳐 조성되는 것을 방지하기 위해 블록(196)에서 테스트가 필요하다. 예를 들면, 워드 "for"의 어간 "fo"는 매우 빈번하게 발생하기 때문에, MinFactor(예를 들면, 1.5의 값)의 적당한 값에 대하여, 블록(195)에

서의 테스트는 공통 워드 "do"가 또한 동일한 노드에서 발생할지라도 만족될 수 있다. 워드 "do"는 분명히 상대적으로 빈번하게 발생되고, "do"가 상기 노드에 대한 디폴트 객체로서 나타나지 않는다면 명확화 시스템의 효율성에 상당한 손실이 될 수 있다.

II 향상된 시스템 특성

1. 구두의 명확성

데이터 키(56; 구두키)의 상부 레프트 키가 구두 문자 마침표, 하이픈, 및 생략부호와 관계되는 본 발명의 바람직한 일 실시 예를 도시한다. 이들 문자 모두 많은 언어, 예를 들면 영어와 불어 모두에 공통적으로 사용된다. 기타 세트의 문자(예를 들면, 문자 콤마, 하이픈 및 긴 대시)가 키에 할당되어 본 발명의 명확한 시스템으로써 상기와 동일한 원리로 접근하여 명확히 할 수 있다. 시스템은 의도하는 구두문자를 결정하기 위해 구두 키(63)상에 키스트로크 주위의 키스트로크의 문맥을 사용한다. 구두키(63)의 디폴트 동작은 임의의 언어로 표현되는 임의의 특정 경우를 조정하기 위해 또한 수정된다.

도 5A와 도 5B에서의 플로우차트는 구두키의 주어진 동작을 발생시키기 위해 시스템이 어떤 문자를 어떻게 결정하는지를 도시한다. 구두키(63)와 관계된 상이한 구두 문자를 사용하여 발생되는 상이한 텍스트 객체가 선택 리스트에 부가된다. 구두키(63)의 시동의 디폴트 해석은 선택 리스트에 있는 제 1 객체로써 부가되는 객체에 의해 결정된다. 대체 해석은 또한 선택키를 활동화 함으로써 또는 디스플레이된 선택 리스트에 있는 소망의 해석을 태평함으로서 선택될 수 있다. 도 5A는 시스템이 키스트로크를 현재의 입력 키스트로크 시퀀스에 부가하는 도 3에 도시된 정규 시스템 처리에 앞서 시스템에 의해 수신된 각 키스트로크를 처리하고, 어떤 워드가 새로운 입력 시퀀스에 대응하는지를 결정하며, 새로운 시퀀스에 근거한 갱신된 선택 리스트를 발생시키는 경로를 도시한다. 도 5A에서 도출되는 처리는 도 3에서의 블록(150A와 175A)에서 발생한다. 도 5B는 새롭게 발생된 선택 리스트를 사후처리하는 경로를 보여준다. 도 5B에 도출된 처리는 도 3의 블록(164B와 178B)에서 발생한다. 모든 플래그는 시스템 초기화 시간에 클리어된다고 가정된다.

각 키스트로크를 사전처리하기 위해 스타트할 때에, 도 5A에 있는 블록(200)에서 플래그 ClearPunctMode가 체크된다. 플래그 ClearPunctMode는 플래그와 버퍼에 관련된 모든 구두키가 다음 키스트로크를 처리하는 처음에서 클리어되는 것을 지시하도록 세팅된다. 블록(200)에서 플래그가 세팅되는 것이 발견되면 다음에 블록(202)에서 구두키(63)를 처리하기 위해 광범위하게 사용되는 모든 플래그와 버퍼가 클리어된다. 이 동일한 플래그와 버퍼는 또한 시스템 초기화 시간에 클리어된다. curPunctWord 버퍼는 구두키 소프트웨어가 선택 리스트에 추가되는 객체를 발생하기 위해 구두문자를 추가하는 텍스트 객체를 세이브하기 위해 사용된다. pState 플래그는 구두키 소프트웨어의 현재 상태를 세이브하고 구두키(3)가 활동화되는 문맥에 대한 상이하고 가능한 경우들 간을 구별한다. 블록(204)에서 현재 사전처리되고 있는 키가 구두키(63)라면, 다음에 블록(206)에서 현재 디폴트(또는 명백히 선택된 객체)가 curPunctWord 버퍼에 세이브되고, 블록(208)에서 플래그 pState는 구두키(63)가 수신된 마지막이라는 것을 지시하는 PUNCT_KEY로 세팅된다. 블록(212)은 블록(151 또는 176)에서 시작하는 도 3에서 도출되는 바와 같이 키스트로크의 정규처리를 나타낸다. 실행은 다음에 커넥터D로부터 시작하는 도 5B에 도시된 사후처리와 함께 계속한다.

블록(204)에서 현재 키가 구두키가 아니라면, 그리고 다음에 블록(220)에서 pState 플래그가 세팅되지 않는다면, 블록(222)에서 구두키를 처리하기 위해 광범위하게 사용되는 모든 플래그와 버퍼가 클리어되고, 키스트로크의 정규 처리가 블록(212)에서 계속된다. 블록(220)에서 pState 플래그가 영이 아닌 값으로 세팅된다면, 그리고 다음에 블록(230)에서 현재 키가 불명확한 데이터 키(56)중의 하나라면, 시스템은 시스템 플래그 Apostrophe_S가 세팅되는지를 테스트하기 위해 블록(232)으로 진행한다.

Apostrophe_S 플래그는 영어와 같은 언어용으로 단지 TRUE를 세팅하며, 워드의 끝에 문자 's'에 이어지는 어포스트로피(소유격을 또는 워드 "is"와 연결하기 위해)를 연결하기 위한 것이다. Apostrophe_S 플래그가 TRUE로 세팅되면, 그리고 다음에 블록(234)에서 pState 플래그가 PUNCT_KEY로 세팅된 것이 발견되면(이전 키 동작이 구두키였다는 것을 나타내는), 그리고 블록(236)에서 현재의 키가 문자 's'와 관련된 데이터 키인것이 발견되면(도 1의 PQRS 키), 다음에 블록(238)에서 pState 플래그는 구두키(63)가 수신되고 's'에 이어지는 특정 경우의 어파스트로피가 인식되는 것을 지시하는 APOS_S로 세팅된다. 테스트(블록(234) 또는 블록(236))가 실패하면 블록(237)에서 pState 플래그는 구두키(63)가 수신되고 구두키 처리가 수행되지만 현재 특정 경우가 적용하지 않는('s'에 이어지는 어파스트로피를 인식하는 것과 같은)것을 지시하는 PUNCT_RCVD로 세팅된다. 어느쪽의 경우에도, pState가 세팅된 후에, 키스트로크의 정규 처리는 블록(212)에서 계속된다.

블록(230)에서, 현재 키가 불명확한 데이터 키가 아닌것으로 결정되면, 블록(250)에서 시스템은 현재 키가 BackSpace 키 인지를 결정한다. 그렇다면, 블록(252)에서, 플래그와 버퍼에 관련된 모든 구두키는 삭제된 키스트로크에 앞서 그 상태로

다시 저장되고 그 후에 키스트로크(BackSpace 키)의 정상 처리는 블록(212)에서 (커넥터C를 통하여) 계속된다. 블록(250)에서 현재 키가 BackSpace 키가 아니라면, 블록(260)에서, 테스트는 현재 키가 모드 키, 선택 키 또는 시프트 키(즉, 현재 워드가 "수용"되지 않고 텍스트 버퍼로 출력되는 키)인지를 결정하기 위해 수행된다. 만일 그렇다면, 키스트로크의 정상 처리는 블록(212)에서 (커넥터C를 통하여) 계속된다. 그렇지 않다면, 블록(260)에서 키는 현재 워드가 수용되도록 하는 키로 결정되고 블록(262)에서 pState 플래그는 클리어되고 ClearPunctMode 플래그가 세팅되어 플래그와 버퍼에 관련된 모든 구두키가 블록(202)에서 다음 키스트로크의 수신하는 즉시 클리어될 것이다.

블록(232)에서, Apostrophe_S 플래그가 세팅되지 않는 것이 결정되면, 블록(270)에서 시스템은 Apostrophe_Term 플래그가 세팅되는지 그리고 블록(272)에서 pState 플래그가 구두키(63)가 수신되고 특정 경우가 선택 리스트에 추가된 어파스트로피에 워드 끝에 인식되었는지를 지시하는 PUNCT_APOS로 세팅되는지를 결정한다. Apostrophe_Term 플래그는 두 개의 워드간에 스페이스를 타이핑하지 않고 이어지는 워드에 어파스트로피로 끝나는 다양한 워드(예를 들면, 1', d', 등)를 다음의 워드와 연결하는 것이 통상적인 불어와 같은 언어용으로 세팅된다. 그러한 언어에서, 사용자가 분명하게 선택키를 활동화하지 않고 이어지는 워드의 타이핑을 즉시 시작하게 하기 위해 함축적인 "선택"을 시스템이 자동적으로 발생하는 것이 매우 편리하다. 블록(270과 272)에서 테스트가 만족된다면, 현재의 디폴트 또는 선택된 워드가 수용되고 마치 선택키가 활동화된 것처럼 텍스트 버퍼로 출력된다. 다음에 블록(276)에서, 현재의 키스트로크 리스트가 클리어되고 실행이 블록(262)으로 진행하며, 그곳에서 pState 플래그가 클리어되고 ClearPunctMode 플래그가 세팅된다. 한편, 블록(270)에서 시스템이 Apostrophe_Term 플래그가 세팅되지 않거나, 또는 블록(272)에서 pState 플래그가 PUNCT_APOS로 세팅되지 않은 것을 결정한다면, 블록(278)에서 pState 플래그는 PUNCT_RCVD로 세팅되고 키스트로크의 정규 처리는 블록(212)에서 (커넥터C를 통하여) 계속된다. 도 5A(그리고 도 3에 상세히 도시됨)의 블록(212)에 의해 나타난 키스트로크의 정상 처리에 이어, 사후처리가 커넥터D로부터 출발하여 도 5B에 도시된 바와 같이 수행된다. 블록(300)에서, pState 플래그가 세팅되지 않는다면, 사후처리가 요구되지 않으며 서브루틴은 단순히 블록(320)으로 복귀한다. 블록(302, 304, 및 306)은 Apostrophe_Term 플래그가 세팅되는지를 체크하며, pState 플래그는 PUNCT_KEY로 세팅되고 현재의 디폴트 워드는 어파스트로피로 끝난다. 세가지 조건 모두 만족한다면, 블록(308)에서 pState 플래그가 PUNCT_APOS로 세팅되고, 그렇지 않으면 시스템은 pState용의 새로운 값으로 세팅하지 않고 블록(310)으로 진행한다. 블록(310)에서 pState 플래그가 PUNCT_APOS 또는 PUNCT_KEY로 세팅된다면, 블록(312, 314, 및 316)에서 curPunctWord 버퍼에 세이브된 텍스트 객체는 마침표, 하이픈, 및 어파스트로피 각각 추가한 후에 선택리스트에 추가된다. 블록(310)에서 테스트가 실패한다면, 그리고 블록(330)에서 pState가 APOS_S로 세팅된 것이 발견된다면, 블록(332)에서 curPunctWord 버퍼에 세이브된 텍스트 객체는 어파스트로피와 's'를 추가한 후에 선택리스트에 부가된다. 다음에 블록(334, 336, 338, 및 340)에서 curPunctWord 버퍼에 세이브된 텍스트 객체는 PQRS키와 관계된 각각의 문자에 뒤따르는 하이픈을 추가한 후에 선택리스트에 추가된다. pState 플래그는 다음에 블록(342)에 있는 PUNCT_RCVD로 세팅된다.

블록(330)에서 pState 플래그가 APOS_S로 세팅되지 않는다면, 블록(352)에서 시스템은 현재 선택 리스트가 비어 있는지를 체크한다. 그렇지 않다면, 하나 이상의 객체는 키스트로크 버퍼(구두키의 적어도 하나의 활동화를 포함하는)에 있는 현재 키 시퀀스를 명백하게 정합하는 데이터베이스에서 발견된다. 이 경우에, 서브루틴은 단순한 키스트로크 버퍼의 문맥을 변화하지 않고 블록(364)으로 복귀하여 데이터베이스는 사용자가 의도한다면 판독될 수 있다. 그렇지 않다면, 데이터베이스에 있는 객체를 더 이상 정합하지 않는 즉시, 블록(356)에서 curPunctWord 버퍼에 세이브된 텍스트 객체는 하이픈이 그것에 추가된 후에 수용된 텍스트로써 출력된다. 블록(358)에서, 키스트로크는 키스트로크 버퍼에서 출발하여 구두키용 키스트로크까지 그리고 그것을 포함하여 삭제된다. 다음에 블록(360)에서 데이터베이스는 질의를 받고 생성된 선택리스트가 수정된 키스트로크 버퍼에 근거하여 발생된다. 마지막으로 블록(362)에서, 구두키용의 모든 버퍼와 플래그는 서브루틴으로부터 복귀하는 것에 앞서 클리어된다.

2. 어휘 모듈 기억 최소화 반면 성능 최대화

도 10에서 도출된 트리 데이터 구조(110)상에서 동작하는 도 13에 도출된 서브루틴(620)의 객체 식별 소프트웨어 처리에 대한 키스트로크의 조합에는 몇개의 새로운 수단이 객체의 더 큰 어휘를 검색하는 반면 서브루틴(620)의 처리시간을 증가하지 않고 어휘 모듈용 기억용량을 더 적게 사용하는 것이 제공된다.

입력 어휘목록에서의 사용의 빈도에 따라 주어진 어휘 모듈의 각 행의 논리 인덱스 테이블(550)에서의 기호의 순서를 정함으로써, 트리 데이터 구조(110)에 있는 모든 노드(400)의 많은 대다수의 명령어(558)는 LOGICAL-SYMBOL-INDEX 필드(555)를 하나와 동일하게 만들 수 있다. 유사하게, 스템 및 워드 객체가 언어에서 사용하는 순서를 감소함으로서 객체 리스트(440)에서 발생될 수 있도록 모든 노드(400)에 있는 모든 패킷형(406)의 명령어(558)를 순서를 정함으로써, 트리 구조(110)에 있는 모든 노드(400)의 많은 대다수의 명령어(558)는 OBJECT-LIST-INDEX 필드(556)를 하나와 동일하게 만들 수 있다. 따라서, 트리(110)에 있는 많은 데이터는 여유가 있다. 따라서, 부모 노드를 자식 노드로 연결하는 경로를 재지정하고 더 이상이 기준이 아닌 자식 노드를 삭제함으로써 대칭적으로 리던던시를 식별하고 그것을 제거하는 것은 거의

노드를 포함하지 않는 래핑된 데이터 구조 또는 고도로 접혀지기 때문에, 오리지널 트리보다 훨씬 적은 명령어, 훨씬 적게 링크하지만 오리지널 트리로부터 검색 가능한 모든 객체를 여전히 검색한다. 더우기, 명령어가 객체 리스트(440)에 있는 유사한 객체를 발생시키는 오리지널 트리를 통한 경로의 분명한 인스턴스는 감소된 구조가 주어진 어휘 모듈의 트리(110)를 정의하기 위해 원시적으로 사용되는 것보다 상당히 더 많은 객체를 발생시키게 하면서 일반화된다(특정에 반대로써) 객체 구조 법칙으로써 기능을 설명하는 접힌 트리에 있는 공통 경로로 나타난다. 예를 들면, 30,000 영어 워드의 리스트로 부터 발생된 접하지 않은 어휘 트리는 바람직한 실시예에서 78,000 명령어 이상을 포함할 수 있다. 폴딩 처리의 바람직한 실시예에 의해 폴딩 후에, 수정 트리는 도 12의 플로우 차트에 있는 바람직한 방법으로 예시된 검색 처리와 주어진 불명확한 키 스트로크 시퀀스를 검색할 수 있는 구조의 워드 객체의 수보다 적은 수의 29,000 명령어보다 더 적게 포함할 수 있다. 각 명령어는 키 누름에 응답하여 단일 기호를 추가함으로써 객체 리스트(430)에 있는 하나의 객체를 변경하기 때문에 놀랍고 새로운 결과가 나타난다. 이것은 폴딩된 트리의 시퀀스와 일반 객체 구조 법칙으로서 명령어의 공통 시퀀스를 다시 사용하는 소프트웨어 처리이다.

도 9에 도출된 노드는 예이다. 노드(560)는 유효 키 필드(562) "010100000"에 있는 "1"에 의해 지시되는 바와 같이 두개의 유효키를 갖는다. 바람직한 실시예에서, "1"의 위치는 2nd와 4th 키가 유효 경로이고 명령어의 패킷과 그것(566과 568)과 관계된 자식 노드에 대한 포인터를 갖는다는 것을 지시한다. 패킷(566)은 자식 노드에 560을 연결하는 포인터 "P"에 의해 이어지는 세개의 명령어 "(1,1,0)", "(1,2,0)", 및 "(2,1,1)"을 포함한다. 서브루틴(600)은 노드(560)로 리딩하는 키스트로크의 리스트를 처리하면, 서브루틴(620)은 "2" 키를 처리하기 위해 호출되고 이것은 이어서 일어나고, 바람직한 실시예에서 ABC 키이다. 명령어(561)는 인덱스 1에 새로운 객체를 구축하기 위해 인덱스 1에서 구 객체에 키의 1st 논리 기호 ABC("a")를 추가한다. 561의 제 3 필드 "0"은 이것이 현재 패킷의 마지막 명령어가 아니라는 것을 지시하는 STOP-FLAG(557)의 거짓값이기 때문에 다음 명령어(563)가 표현된다. 명령어(563)는 인덱스(2)에서 새로운 객체를 구축하기 위해 1st 논리 기호 키 ABC("a")를 인덱스(2)에서 구 객체에 추가한다. 구축된 새로운 객체의 인덱스는 명령어 자체의 순서내에서 목시적이기 때문에, 예를들면, 2nd명령어는 항상 2nd객체를 구축하기 때문에, 명령어 새로운 객체의 인덱스는 2가 될 것이다. 명령어(563)의 제 3 필드("0")는 STOP-FLAG(557)의 거짓값이기 때문에, 다음 명령어(567)가 해석된다. 명령어(567)는 인덱스(3)에서 새로운 객체를 구축하기 위해, 키(ABC(a"c"))의 2nd논리 기호를 인덱스(1)에 있는 구 객체에 추가한다. 명령어(567)의 제 3 필드(1)는 STOP-FLAG(557)의 참값이면서, 이것이 현재 패킷의 최종 명령어인가를 나타내기 때문에, 서브루틴(620)의 실행은 블록(638)에서 블록(644)으로 통과할 것이다.

다른 명령어 패킷을 포함하는 2개 이상의 노드를, 동일한 목적을 개별적인 다중 노드로서 제공할 수 있는 단일노드로 결합하는 것이 가능하며, 이것은 어휘 트리(110)에 있는 어떤 노드가 새로운 센스에서 중복성이 있음을 의미한다. 본발며의 목적을 위하여, 워드 "중복성"은 한 노드가, 도 14-17에 있는 바람직한 실시예에서 설명된 소프트웨어 프로세스의 동작에 의해 시행될 수 있는 센스에 있는 2개의 노드에 대하여 사용된다.

예를들면, 도 9에 있는 노드(560)를 노드(574)로 비교하라. 키(2)상에 있는 명령어 패킷(566,571)은 정확하게 일치하지만, 노드(560)의 키(4)상에 있는 명령어(570)는 노드(574)의 키(4)상에 있는 명령어(572)와 대립하기 때문에, 서로의 작업을 할 수 없으며, 2개의 노드가 서로의 작업을 하는 하나의 노드로 결합할 수 없다. 노드(560)을 노드(576)에 비교하라. 각 노드상에 있는 키(2)와 관련된 명령어 패킷(566,577)은 정확하게 일치한다. 명령어(569,578)는 그들의 STOP-FLAG 필드(557)의 세팅에 있어 다르지만, 이들 차이는 이들이 대립하게는 만들지 않는다. 도 14에 있는 서브루틴(620)의 객체 검색공정의 본질적인 결과는 주어진 키에 대한 노드에서 명령어 세트를 실행함으로써 만들어진다. 부가적인 객체는 임의의 노드 자식의 올바른 프로세싱에 손해없이, 최종 객체리스트에 추가될 수 있다. 따라서, 578뒤의 추가 명령어의 실행에 의해 노드(576) 자식의 프로세싱에서 에러가 생기지 않는다. 단지 올바르지 않은 명령어가 실행될 가능성이 있었거나, 너무 적은 명령어가 실행될 가능성이 있었다면, 필수 공정은 교란되었을 것이다. 동일하게, 노드(576)의 키(9)상에 있는 유효키의 존재는 노드(560)상에 있는 유효(9)키의 부재와 대립되지 않는다. 따라서, 노드(560,576)는 중복성이 있으며, 양쪽의 순효과를 이루며, 양쪽의 자식의 부모노드로서 정확한 기능을 하는 새로운 노드(582)로 합병된다. 또한 지시기는 중복성을 정의하는 역할을 한다. 더 긴 워드의 어간을 계속 형성하지 않는 워드와 관련된 트리에 있는 최종 키스트로크 시퀀스에서, 유효 키 패킷(408)에 있는 지시기는 바람직한 실시예에서 추가의 자식이 없는 것을 지시하는 특정 값(NULL)을 가진다. 상기 노드를 "터미널 노드"로 명명한다. 양쪽 노드에 공통적으로 유효 키상에 있는 자식 노드를 갖는 2개의 노드에 대해, 터미널 노드가 도착할 때까지 또는 비교되고 있는 노드에 공통적인 유효 키 시퀀스상에 더이상 내림차순이 없을 때까지, 각각의 자식노드는 자식으로부터 내림차순한 노드에 중복성 등이 있기 위해서, 그들의 부모노드에 대해 중복성이 있어야 한다.

도 14 내지 17은 도 10에 도시된 트리(110)에 유사한 어휘 모듈 트리의 폴딩 및 압축을 위한 소프트웨어 프로세스의 바람직한 실시예의 플로차트를 설명한다. 도 14는 압축된 어휘 모듈을 구축하기 위한 소프트웨어 프로세스의 바람직한 실시예

의 플로차트이다. 블록(652)에서, 프랑스 어휘 모듈을 위한 도 6에서와 같이, 물리적인 키상에 나타나는 이들 이외의 다른 임의의 필수적인 추가의 모호한 기호를 식별하기 위해서, 어휘사전이 스캐닝된다. 블록(654-656)에서, 기호는 도 7의 실시예에서와 같이, 입력된 어휘사전에 있는 사용회수의 감소순서로 기호의 각각의 키상에 있는 그들의 논리적 인덱스가 부여된다.

회수를 갖는 객체의 어휘사전이 주어질 때, 블록(658)이 형식(110)의 어휘트리를 어떻게 구축하는지는 당업자는 잘 알것이다. 블록(660)에서, 중복 노드가 식별되며, 데이터의 복사를 최소화하기 위해 서로 합병되며, 따라서, 단일 객체와 관련된 분리된 명령어 시퀀스를 다중 객체를 검색하는 일반화된 룰로 변하게 한다. 이러한 공정은 도 15에 상세히 도시되어 있다. 블록(662)은 모든 잔존 NULL 지시기를 터미널 노드로부터 식별하며, 가장 큰수의 부모를 가진 노드에 지시하기 위해 이들을 바꾸며, 따라서, 모듈에 있는 룰의 수를 증가시킨다. 다른 룰은 할당한 자식노드에 NULL 지시기로 적용할 수 있으며, 이러한 룰은 프로세스되는 키스트로크에 관련된 인자를 근거로 하여, 객체가 검색될 때, 다이내믹하게 적용될 수 있다. 블록(664)에서, 각 고유 명령어(558) 및 지시기(404a)의 잔존 인스턴스는 카운트되어, 이들은 스페이스를 보존하기 위해, 더 높은 회수 명령어 및 주소에 할당된 더 짧은 비트패턴을 갖는 비트의 고유 패턴으로서 암호화될 수 있다. 본 발명의 바람직한 실시예에서, Huffman 코딩은 최소길이 비트 패턴을 명령어 및 주소에 할당하기 위해 사용된다. Huffman 코딩 기술은 종래기술로 잘 알려져 있으며, 따라서, 여기서는 상세히 설명할 필요가 없다. 더욱기, 다중 부모 노드의 자식 노드가 되는 노드는 그들의 빠른 검색을 용이하게 하며, 이들을 주소화하기 위해 요구되는 비트의 수를 최소화하기 위해서, 특정 순서로 저장될 수 있다. 마지막으로, 위에서 설명된 바와 같이 생성된 데이터는 블록(666)에 있는 파일로 기록된다.

블록(658)에서 트리를 구축하면서 저장되어야 할 객체를 설명하기 위해 명령(558)을 선택하는 데 있어서, 객체가 워드 또는 워드의 어간일 때 워드의 문자 시퀀스는 트리(110)의 노드 리던던시를 유익하게 증가시키는 데 사용될 수 있는 추가 데이터를 포함하는 것이 인식될 것이다. 예로서 영어의 모든 글자 쌍이 전부 마찬가지로 공통인 것은 아닌 데, 예를들면 "s"는 "t"와 쌍을 이룬다. 글자쌍 또는 바이그램의 통계치가 이전 글자로부터 객체의 다음 글자일 가능성성이 가장 높은 글자를 예측하는 데 사용될 수 있다. 이러한 예측으로 논리 기호 인덱스 테이블(550)에서 모호한 기호의 논리적 순서화가 제 1위치의 사용을 더욱 최적화하기 위해 동적으로 변경될 수 있다. 이 예측은 글자 트리플렛, 트리그램 및 일반적으로는 n-그램까지 확장될 수 있다.

도 15는 어휘 모듈의 트리(110)를 폴딩시키기 위한 소프트웨어 프로세스의 바람직한 실시예인 흐름도이다. 블록(670)은 도 14의 흐름도의 블록(660)으로부터 이어진다. 블록(672)은 루트 노드(111) 다음에 도 10의 트리(110)의 첫번째 노드로 프로세스를 초기화한다. 블록(674)은 현재 노드와 함께 최대로 여분이 있는 노드를 위치시키기 위해 도 16의 흐름도의 바람직한 실시예에 나타낸 서브루틴을 호출한다. 목적지 노드가 발견되면, 결정 블록(676)은 여분 노드가 함께 병합되는 블록(678)으로 프로세스가 진행되게하고, 복수 개의 개별 명령 사례들을 공유 시퀀스로 차위 합계하는 트리로부터의 데이터 중복을 제거하며, 이것들은 키스트로크 시퀀스를 객체에 연관시키는 일반적인 규칙이다. 결정 블록(676)에서 오류가 발생하면, 판정 블록(680)은 프로세스가 수행완료되었는 지의 여부를 결정한다. 처리하여야 할 노드가 더 있으면, 흐름은 다른 노드를 식별하기 위해 블록(682)으로 간다.

도 16은 주어진 노드에 대해 최대 리던던시 정도를 갖는 노드를 트리(110)에서 발견하기 위한 소프트웨어 프로세스의 바람직한 실시예의 흐름도이다. 블록(690)은 도 15의 흐름도의 블록(674)으로부터 이어진다. 블록(692)은 측정된 리던던시를 위한 플레이스홀더인 MAX-SAVINGS를 초기화한다. 블록(694)은 도 10의 트리(110)의 루트 노드(11)에서 시작하도록 프로세스를 초기화한다. 블록(696)은 현재 노드에 대해 주어진 노드의 리던던시를 계산하기 위해 도 17의 바람직한 실시예의 흐름도에 나타낸 서브루틴(710)을 호출한다. 판정 블록(698)은 MAX-SAVINGS 보다 큰 리던던시 정도가 보고되었는 지의 여부를 테스트한다. 만일 그렇다면, 블록(700)은 주어진 노드에 대해 가장 리던던트한 것으로 일시적으로 발견된 신분 BEST-NODE와 MAX-SAVINGS으로서 보고된 리던던시 측정치를 기록한다. 판정 블록(702)은 모든 노드가 평가되었는 지의 여부를 테스트한다. 만일 그렇지 않다면, 흐름은 블록(704)으로 계속하고, 현재 노드로부터 다음 노드로 진행한다. 판정 블록(702)에서의 결과가 최종 노드가 평가된 것으로 나타난다면, 블록(706)은 주어진 노드에 대해 최대 리던던트인 노드의 신분을 도 15의 서브루틴(670)의 블록(674)에 반환한다.

도 17은 두 특정 노드간의 수치적 리던던시를 계산하기 위한 소프트웨어 프로세스의 바람직한 실시예의 흐름도이다. 블록(710)은 도 16의 서브루틴(690)의 흐름도의 블록(696)으로부터 이어진다. 블록(712)은 중복 명령의 수를 초기화한다. 블록(714)은 KEY-INDEX를 1로 초기화한다. 블록(716)은 서브루틴에 대한 파라미터로서 특정된 두 노드(400)중 첫번째로부터 KEY-INDEX와 연관된 명령 패킷(406)을 판독하여 임시 리스트인 LIST-A에 임시로 넣는다. KEY-INDEX가 유효키가 아니면, 어떠한 명령도 판독되지 않는다. 블록(718)은 서브루틴에 대한 파라미터로서 특정된 두 노드(400)중 두번째로부터 KEY-INDEX와 연관된 명령 패킷(406)을 판독하여 임시 리스트인 LIST-B에 임시로 넣는다. KEY-INDEX가 유효키가 아니면, 어떠한 명령도 판독되지 않는다. 판정 블록(720)은 LIST-A 또는 LIST-B가 비워졌는 지의 여부를 결정한다. 만일 그렇지 않다면, 블록(722)은 LIST-A 및 LIST-B로부터 하나의 명령을 인출하고, 각각에 남아있는 명령의 수를 하나

씩 감소시킨다. 판정 블록(724)은 명령이 LOGICAL-SYMBOL-INDEX 와 OBJECT-LIST-INDEX 필드에서 동일한 지의 여부를 테스트한다. 만일 그렇지않다면, 어떠한 리턴던시도 없다는 오류코드가 블록(726)에서 서브루틴(690)의 블록(696)으로 반환된다. 판정 블록(724)의 결과가 '예'이면, 블록(728)은 SAVED-INSTRUCTION 수를 증가시킨다. 제어는 블록(720)으로 전달된다. 판정블록(720)이 TRUE를 테스트하면, 제어는 판정블록(730)으로 가고, 두 노드가 모든 가능한 키에 대해 비교되었는지의 여부를 테스트한다. 만일 그렇지않다면, 블록(730)은 KEY-INDEX를 증가시키고, 제어는 판정 블록(716)으로 간다. 블록(730)에서의 결정이 긍정이면, 제어는 KEY-INDEX를 1로 리셋팅시키기 위해 블록(734)으로 간다. 블록(736)은 두 노드의 키 KEY-INDEX와 연관된 포인터를 검사한다. 판정블록(738)은 포인터가 종료 노드에서의 포인터인 경우 또는 비유효 키에 대한 경우일 수 있는 앱프티(NULL)인 지의 여부를 테스트한다. 어떠한 포인터도 앱프티 상태가 아니면, 제어는 블록(740)으로 가고, 이것은 두 개의 비-애프티 포인터에 의해 포인팅된 자식 노드가 리턴던트 상태인 지의 여부를 재귀적으로 테스트하기 위해 서브루틴을 이용한다. 블록(740)의 결과는 판정블록(742)에서 판정된다. 두 자식 노드가 리턴던트 상태가 아닌 것으로 판정된다면, 오류코드는 블록(744)에 반환된다. 그렇지않으면, 두 자식은 블록(746)에서 측정된 일정한 수치적 점수를 갖는 리턴던트인 것으로 발견된다. 판정블록(748)은 최종 키(바람직한 실시예에서 키 9)와 연관된 포인터가 테스트되었는지의 여부를 테스트한다. 만일 그렇지않다면, 블록(752)은 KEY-INDEX를 증가시키고 제어를 블록(736)으로 전달한다. 블록(748)에서의 테스트가 모든 포인터가 검사된 것으로 결정한다면, 서브루틴이 블록(710)에 호출되어 왔을 때 최초로 식별된 두 노드의 측정된 리턴던시 수치 측정치가 반환된다.

리턴던시의 수치값 계산은 자식으로서 노드에 포인팅하는 부모의 수와 각 노드에 있는 가지의 수와 같은 추가 인자를 고려하기 위해 가중될 수 있음이 인식될 것이다. 두 노드가 일정 키와 연관된 명령의 순서화로 인해 리턴던트 상태가 아니라면, 입력 용어집에서 저빈도 단어와 연관된 명령의 순은 고빈도 객체와 연관된 명령의 우선순위에 영향을 미치지 않고 재순서화될 수 있고, 이에따라 트리의 리턴던시를 증가시키는 것이 인식될 것이다.

III. 대표적 시스템 동작

도 18a 내지 18k는 감소된 키보드 명확화 시스템의 대표적 사용 동안 휴대용 컴퓨터(52)의 디스플레이(53)를 나타낸다. 휴대용 컴퓨터(52)의 전원을 켠 후엔, 텍스트 영역(66) 및 선택 리스트 영역(70)은 빈 상태로 된다. 도 18a에서, 사용자는 문장 "This is a test"라는 문장을 타이핑하였다. 어휘 모듈은 단어"test"와 같은 최종 4개의 키스트로크(TUV,DEF,PQRS,TUV)의 시퀀스를 가장 가능성있는 해석으로 식별하고 이 해석을 삽입 포인트(900)에 임시로 위치시킨 후 이 단어를 먼저 디폴트 해석(901)으로서 선택 리스트에 놓는다. 선택 리스트에서 이 첫번째 엔트리는 그 둘레의 실선 박스로 도시되었고(이것은 목시적으로 선택된 객체를 지시함), 그 둘레의 실선 박스로 도시된 삽입 포인트(900)에서 텍스트 영역에 임시로 위치된다. 어휘 모듈은 단어 해석 "vest" 와 어간해석 "vert"를 식별하여, 이들 해석을 선택 리스트에 놓는다. 도 5a 및 5b에 도시된 구두점 키 프로세싱 소프트웨어는 그 curPunctWord 버퍼에 텍스트 객체 "test"를 저장하였다.

도 18b에서, 사용자는 순차적으로 .-' 구두점 키(63)를 눌렀다. 데이터베이스의 어떠한 객체도 구두점 키(63)가 뒤따르는 4개의 키(TUV,DEF,PQRS,TUV)의 키스트로크의 시퀀스와 매칭되지 않는다. 따라서, 선택 리스트에 나타나는 텍스트 객체만이 첨부되는 마침표(911), 하이픈(913) 및 어포스트로피(914)를 갖춘 curPunctWord "test"로 이루어 진, 구두점 키 프로세싱 소프트웨어에 의해 부가된 것들이다. 최종 디폴트 선택 리스트 객체는 "test"(911)이고, 이것은 삽입 포인트(910)에서 텍스트 영역에 임시로 위치된다.

도 18c에서, 사용자는 후속하여 스페이스 키(922)를 눌렀다. 이것은 명시적인 스페이스 문자를 발생시켰고, 현재 키스트로크 시퀀스를 종료시키고 시스템이 임시로 위치된 텍스트 "test"를 수용하게 하였다. 다음 스페이스도 시스템 커서(920)가 나타난 후 출력되었다. 키스트로크 버퍼는 이제 빈 상태이고, 따라서 현재 선택 리스트(921) 역시 빈 상태이다. 이것은 스페이스(또는 선택) 키의 작동 이전에 작동될 때 구두점 키(63)의 디폴트 작용중의 하나를 나타내고, 대응하는 구두점 문자를 포함하는 데이터베이스에서 객체와의 매칭이 없는 경우에, 이전 디폴트 텍스트 객체가 첨부된 마침표를 갖춰 출력된다.

영어(여기서 플래그 Apostrophe_S는 TRUE로 설정됨)를 위해 디자인된 시스템의 경우에, 도 18d는 사용자가 PQRS 키(932)를 눌렀을 때 도 18b를 따르는 결과를 나타낸다. 이것은 도 5B의 블록(332) 내지 블록(342)에 나타난 프로세싱을 트리거링하고, 구두점 키 프로세싱 소프트웨어는 객체 "test's", "test-s", "test-r", "test-p" 및 "test-q"를 상기 순서대로 선택리스트에 부가한다. 따라서, 최종 디폴트 선택 리스트 객체는 "test's"(931)이고, 이것은 또한 삽입 포인트(930)에서 텍스트 영역에 임시로 위치된다.

도 18e는 사용자가 TUV 키(932)를 눌렀을 때(하이픈이 있는 단어"test-run"를 타이핑했다는 생각과 함께) 도 18d의 결과를 따른다. 이것은 도 5B의 블록(332) 내지 블록(340)에 도시된 프로세싱을 트리거링하고, 구두점 키 프로세싱 소프트웨어가 텍스트 객체 "test-"가 수용된 텍스트로서 출력되어지게 하고, 키스트로크 버퍼가 단지 최종 두 키스트로크(PQRS

및 TUV)만이 포함되도록 수정되어지게 하는 결과로 된다. 이들 두 키스트로크에 기초한 데이터베이스에 대한 질의(도 5B의 블록(360))는 도 18e에 도시된 선택 리스트가 되는 결과로 된다. 텍스트 객체 "st"는 최빈도로 나타나므로 디폴트 선택 리스트 객체 "st"(941)로서 나타나고, 역시 삽입 포인트(940)에서 텍스트 영역에 임시로 위치된다.

도 18f는 사용자가 하이픈이 있는 단어 "test-run"의 타이핑을 종료하기 위해 MNO 키(952)를 눌렀을 때 도 18e를 따르는 결과를 나타낸다. 어떠한 특정 구두점 키 프로세싱도 모든 플래그가 이전 키스트로크의 처리에서 도 5B의 블록(362)에서 클리어 되었으므로 발생하지 않는다. 따라서, 시스템은 단순히 현재 키 시퀀스 PQRS,TUV,MNO를 처리한다. 이들 3개 키스트로크에 기초한 데이터베이스를 질의하는 것은 도 18f에 도시된 선택 리스트로 된다. 텍스트 객체 "run"는 최빈으로 나타나고 따라서 디폴트 선택 리스트 객체 "run"(951)으로서 나타나고, 역시 삽입 포인트(950)에서 텍스트 영역에 임시로 위치된다.

도 18g 내지 18k는 프랑스어(여기서 플래그 Apostrophe_term은 TRUE로 설정됨)를 위해 디자인된 시스템의 예를 나타낸다. 본 예에서, 단어 "est"와 연결된 단어 "c"로 이루어지는 텍스트 객체 "c'est"를 타이핑한다. 도 18g는 사용자가 ABC 키(962)를 눌렀을 때의 결과를 나타낸다. 선택 리스트는 액센트가 있는 문자인 à, â, 및 ç를 포함하는, 프랑스어에 대한 ABC 키와 연관된 단일 문자 객체를 도시한다. 어떠한 특수 구두점 키 프로세싱도 모든 구두점 키 관련 플래그가 이전에 클리어되었으므로 발생하지 않는다. 시스템은 단순히 현재 단일-키 시퀀스 ABC를 처리한다. 이들 키스트로크에 기초한 데이터베이스 질의는 도 18g에 도시된 선택 리스트로 된다. 텍스트 객체 "à"는 최빈으로 나타나고 디폴트 선택 리스트 객체 "à"(961)으로서 나타나고, 역시 삽입 포인트(960)에서 텍스트 영역에 임시로 위치된다.

도 18h에서, 사용자는 후속하여 .-' 구두점 키(63)를 눌렀다. 객체 "c'" 및 "a-"는 데이터베이스에 저장되고, 데이터베이스에 질의한 후 도 3의 블록(156) 내지 블록(160)에서 선택 리스트에 추가되고, 따라서 최종 디폴트 선택 리스트 객체는 "c'"(971)이다. 결과적으로, 도 5B의 블록(300) 내지 블록(306)에서 각각의 테스트가 연속하여 수행되고, pState 플래그는 블록(308)에서 PUNCT_APOS로 설정된다. 선택 리스트에서 객체 "c'"(971) 및 "a-"(973)에 뒤이어, 구두점 키 프로세싱 소프트웨어(도 5B의 블록(312,314 및 316))는 첨부된 마침표(974), 하이픈(975), 및 어포스트로피(976)가 있는 curPunctWord "à"로 이루어진 객체 "à.", "à-", 및 "à'"를 추가한다. 최종 디폴트 선택 리스트 객체는 "c'"(971)이고, 이것은 또한 삽입 포인트(970)에서 텍스트 영역에 임시로 위치된다.

도 18i에서, 사용자는 후속하여 DEF키(982)를 눌렀다. pState 플래그는 도 18h에서 이전 키스트로크 프로세싱의 결과로서 PUNCT_APOS로 설정되었으므로, 구두점 키 프로세싱 소프트웨어는 도 5A에 블록(274,276 및 262)을 실행한다. 현재 디폴트 선택 리스트 객체 "c'"는 수용된 텍스트(983)로서 출력되고, 현재 키스트로크 버퍼는 클리어되며, 시스템은 단단히 현재 키스트로크 DEF를 클리어된 키스트로크 버퍼에 추가하는 결과로서 최종 단일-키 시퀀스 DEF를 처리한다. 이들 키스트로크에 기초한 데이터베이스 질의는 도 18i에 도시된 선택 리스트가 되는 결과가 된다. 텍스트 객체 "d"는 디폴트 선택 리스트 객체 "d"(981)로서 최빈으로 나타나고, 삽입 포인트(980)에서 임시로 텍스트 영역에 위치된다.

도 18j 및 18k는 사용자가 단어 "est" 타이핑을 종료하기 위해 PQRS 키 (992)(도 18j에서) 및 TUV 키(997)(도 18k에서)를 누를 때 도 18i를 따르는 결과를 나타낸다. ClearPunctMode 플래그가 이전 키스트로크의 처리에서 도 5A의 블록(262)에서 설정되었으므로 어떠한 특정 구두점 키 프로세싱도 일어나지 않으며, 따라서 구두점 키를 위한 모든 버퍼 및 플래그는 도 5A의 블록(202)에서 클리어된다. 따라서 시스템은 단순히 현재 키 시퀀스 DEF,PQRS,TUV를 처리한다. 이들 키스트로크에 기초한 데이터베이스 질의는 도 18k에 도시된 선택 리스트가 되는 결과가 된다. 텍스트 객체 "est"는 디폴트 선택 리스트 객체 "est"(996)로서 최빈으로 나타나고, 삽입 포인트(995)에서 임시로 텍스트 영역에 위치된다.

프랑스어는 le,ne,ce,de,me,se,je,que,tu 및 te와 같은 다수의 빈번하게 사용되는 대명사, 전치사 및 접두사를 포함하며, 모음으로 시작하는 단어에 선행할 때, 그 최종 모음은 어포스트로피로 변화되고 다음 단어 이전에 어떠한 공간도 없이 타이핑된다. 상기 예에 도시된 구두점 키(63)에 의해 지원되는 작용없이, 단어 객체의 이와 같은 빈번한 연결(concatenation)은 사용자로 하여금 두번째 단어 객체(est) 이전에 맨처음 단어 객체(도시된 예에서 c')를 종료하기 위해 선택 키의 추가적인 키 활성화를 요구할 것이다. 따라서, 어포스트로피로 끝나는 단어 객체에 자동 연결을 지원하는 구두점 키(63)의 성능은 프랑스어와 같은 언어를 위한 시스템의 자연적인 느낌과 효율을 증대시킨다.

본 발명의 바람직한 실시예가 예시되고 설명되었지만, 본 발명의 범위 및 정신으로부터 벗어나지 않고 다양한 변경이 있을 수 있음이 인식될 것이다. 예로서, 당업자는 감소된 키보드 명확화 시스템의 키보드(54)가 적게는 3개 또는 많게는 20개 데이터 키를 가질 수 있음이 인식될 것이다. 본 명세서에 개시된 명확화 기술은 상이한 사이즈의 키보드에도 마찬가지로 적용될 수 있다. 구두점 키(63)와 연관된 특정 문자는 변경될 수 있다. 예로서, 당업자는 마침표가 일정 응용의 경우엔 콤마

로 대체될 수 있음을 인식할 것이다. 또한, 상이한 주변 키스트로크 문맥에서 발생된 특정 선택사항의 구두점 문자도 변경될 수 있다. 예로서, 구두점 키(63)의 두 개의 연속하는 활성화에는 엔 대쉬'-'가 발생하는 결과가 뒤따르는 반면에 세 개의 연속하는 활성화에는 생략부호인 "..."가 발생할 수 있다.

당업자는 또한 추가적인 어휘 모듈 예로서, 법률 용어, 의학 용어 및 외국어 용어를 포함하는 어휘모듈이 컴퓨터에서 인에 이블될 수 있음을 인식할 것이다. 시스템 메뉴를 통해, 사용자는 추가의 어휘 단어가 특정한 컬러 또는 하이라이트되게 함과 함께, 가능한 단어 리스트의 맨처음 또는 맨나중에 나타날 수 있도록 구성할 수 있다. 결과적으로, 본 발명은 본 명세서에서 특정하게 설명된 실시예 보단 첨부된 특허청구의 범위내에서, 실시될 수 있음이 인식되어야 할 것이다.

(57) 청구의 범위

청구항 1.

사용자에 의해 입력된 모호한 입력 시퀀스를 명확하게 하고,

- (a) 복수의 문자와 각각 관련된 복수의 입력을 가진 사용자 입력 디바이스;
- (b) 입력 시퀀스 및 상대적인 사용 빈도수와 각각 관련된 복수의 객체를 구성하는데 사용되는 데이터를 포함하는 메모리;
- (c) 사용자에게 시스템 출력을 묘사하는 디스플레이; 및
- (d) 상기 사용자 입력 디바이스, 메모리, 및 디스플레이에 연결된 프로세서;를 포함하며,

선택된 입력의 연속 동작에 의해 생성된 입력 시퀀스는 각각의 입력과 연관된 복수의 문자로 인해 모호한 해석을 가지며, 모든 객체는 복수의 노드로 구성된 트리 구조로 메모리에 저장되어 있고, 각각의 노드는 입력 시퀀스 및 하나이상의 객체와 연관되어 있고, 복수의 노드는 복수의 경로에 의해 연결되어 있고, 복수의 경로의 각각은 베이스 입력 시퀀스와 연관된 모 노드를 추가 입력 및 상기 모 노드의 베이스 입력 시퀀스와 연관된 자 노드와 링크하고, 자 노드와 연관된 객체는 자 노드가 링크되어 있는 대응 모 노드와 연관된 객체를 기초로 하고, 대응 모 노드와 연관된 객체를 수정하기 위해 메모리에 사전 저장된 코드를 이용하여 구성되고, 그 코드는 대응 모 노드와 연관된 객체의 숫자 색인의 명세와 모 노드를 자 노드에 링크하는 추가 입력과 연관된 문자 중 하나의 숫자 색인의 명세를 포함하는 대응 모 노드와 연관된 객체를 수정함으로써 자 노드와 연관된 객체를 구성하는데 사용되고, 상기 프로세서는:

(i) 각각의 생성된 입력 시퀀스와 연관된 메모리내의 하나이상의 객체로부터, 가장 많은 사용 빈도수를 가지며 적어도 하나의 워드 객체가 생성된 입력 시퀀스와 연관되어 있을 때 워드 객체이고 생성된 입력 시퀀스와 연관된 워드 객체가 없을 때 워드 스템 객체인 적어도 하나의 후보 객체를 식별하고;

(ii) 디스플레이로 하여금 상기 생성된 입력 시퀀스의 원문 해석으로서 각각의 생성된 입력 시퀀스와 연관된 적어도 하나의 식별된 후보 객체를 디스플레이하게 하는 출력 신호를 생성하는 명확화 시스템에 있어서,

상기 메모리 트리 구조는 중복 모 노드를 제거하도록 배열되어 있고, 트리의 두 개의 모 노드는 두 개의 모 노드 모두에 존재하는 주어진 입력 시퀀스와 연관된 모든 코드가 그 코드가 동일 시퀀스에서 발생하고 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정한다는 점에서 동일할 때 중복되어 있고, 추가로, 자 노드가 두 개의 모 노드의 각각에 링크되어 있는 모든 입력에 대하여, 상기 자 노드는 동일 귀납 센스에서 또한 중복하고, 상기 중복 모 노드 중 하나는 메모리의 트리 구조로부터 생략되어 있고, 남은 중복 모 노드는 생략된 중복 모 노드에만 존재하였던 자 노드로의 링크 및 임의의 코드에 의해 증대되고, 생략된 모 노드가 링크로서 재기록되는 자 노드인 임의의 모 노드로부터 남은 중복 모 노드로 모두 링크하는 것을 특징으로 하는 시스템.

청구항 2.

제 1 항에 있어서, 특수 발음 구별 부호를 포함하는 하나이상의 문자는 특수 발음 구별 부호가 없는 대응 문자가 연관되어 있는 동일 입력과 연관되어 있는 것을 특징으로 하는 시스템.

청구항 3.

제 2 항에 있어서, 특수 발음 구별 부호를 포함하는 하나이상의 문자는 특수 발음 구별 부호가 없는 대응 문자와 연관된 입력에 나타나지 않는 것을 특징으로 하는 시스템.

청구항 4.

제 1 항에 있어서, 상기 생성된 입력 시퀀스의 원문 해석으로서 디스플레이되는 최고 사용 빈도수의 적어도 하나의 식별된 후보 객체는 상기 워드 스템 객체의 사용 빈도수가 상기 생성된 입력 시퀀스와 연관된 임의의 워드 객체와 연관된 최고 사용 빈도수를 소정 비율만큼 초과할 때 워드 스템 객체이고, 상기 생성된 입력 시퀀스와 연관된 임의의 워드 객체와 연관된 상기 최고 사용 빈도수는 소정 한계치를 초과하지 않는 것을 특징으로 하는 시스템.

청구항 5.

제 1 항에 있어서, 상기 생성된 입력 시퀀스의 원문 해석으로서 임시로 디스플레이되는 최고 사용 빈도수의 적어도 하나의 객체는 하나이상의 문자의 모호하지 않은 생성을 야기하는 임의의 입력을 입력하는 즉시 상기 생성된 입력 시퀀스의 원문 해석으로서 출력에 대해 확인되고, 상기 하나이상의 모호하지 않은 문자는 상기 확인된 원문 해석의 출력에 바로 후속하여 출력되는 것을 특징으로 하는 시스템.

청구항 6.

제 1 항에 있어서, 대응 모 노드와 연관된 객체를 수정함으로써 자 노드와 연관된 객체를 구성하는데 사용되는 코드는 상기 코드가 자 노드와 연관된 객체를 만드는 코드 시퀀스의 최종 코드인지에 대한 명세를 더 포함하는 것을 특징으로 하는 시스템.

청구항 7.

제 1 항에 있어서, 모 노드에 링크된 자 노드에 대응하는 추가 입력의 수와 일치성은 상기 자 노드의 수와 일치성을 표시하는 유효 키 비트의 필드에 의해 모 노드에 표시되는 것을 특징으로 하는 시스템.

청구항 8.

제 7 항에 있어서, 자 노드로의 포인터는 자 노드와 연관된 객체를 만드는데 사용되는 하나이상의 코드의 각각의 세트에 바로 이어지고, 하나이상의 코드의 하나이상의 세트와 이후의 포인터는 상기 자 노드의 수와 일치성을 표시하는 유효 키 비트와 동일 순서로 모 노드내의 메모리에 순차적으로 배치되어 있는 것을 특징으로 하는 시스템.

청구항 9.

제 1 항에 있어서, 자 노드와 연관된 객체를 만드는 코드 시퀀스는 객체가 상기 객체의 사용 빈도수와 관련하여 분류되어 있는 시퀀스로 생성되도록 메모리에 정렬되어 있는 것을 특징으로 하는 시스템.

청구항 10.

제 1 항에 있어서, 각각의 입력과 연관된 문자의 색인은 메모리내의 워드 객체의 문자 발생 빈도수의 하향 순서로 문자에 순차적으로 할당되어 있는 것을 특징으로 하는 시스템.

청구항 11.

제 1 항에 있어서, 각각의 입력과 관련된 문자의 색인은 복수의 문자를 구성하는 스트링에 할당되어 있는 색인을 포함하는 것을 특징으로 하는 시스템.

청구항 12.

제 1 항에 있어서, 대응 모 노드와 연관된 객체를 수정함으로써 자 노드와 연관된 객체를 구성하는데 사용되는 코드는 자 노드와 연관된 구성 객체와 연관된 객체 타입의 명세를 더 포함하는 것을 특징으로 하는 시스템.

청구항 13.

제 12 항에 있어서, 지정되는 상기 객체 타입은 구성 객체의 언어 부분에 관한 정보를 포함하는 것을 특징으로 하는 시스템.

청구항 14.

제 12 항에 있어서, 지정되는 상기 객체 타입은 구성 객체의 대문자 사용에 관한 정보를 포함하는 것을 특징으로 하는 시스템.

청구항 15.

제 12 항에 있어서, 지정되는 상기 객체 타입은 구성 객체에 첨부될 수 있는 어형 변화 어미와 접미사에 관한 정보를 포함하는 것을 특징으로 하는 시스템.

청구항 16.

제 12 항에 있어서, 지정되는 상기 객체 타입은 메모리내의 객체 중 상기 구성 객체를 유일하게 식별하는 코드를 포함하는 것을 특징으로 하는 시스템.

청구항 17.

제 12 항에 있어서, 지정되는 상기 객체 타입은 상기 구성 객체의 사용 빈도수에 관한 정보를 포함하는 것을 특징으로 하는 시스템.

청구항 18.

제 12 항에 있어서, 지정되는 상기 객체는 상기 구성 객체가 완료 워드인지에 관한 정보를 포함하는 것을 특징으로 하는 시스템.

청구항 19.

제 1 항에 있어서, 각각의 입력과 관련된 문자의 색인은 상기 색인 문자가 자 노드와 관련된 객체를 형성하는데 부가되는 대응 모 노드와 관련된 객체에서 바로 앞선 문자에 후속하는 문자의 발생 빈도수의 하향 순서로 문자에 순서적으로 할당되어 있는 것을 특징으로 하는 시스템.

청구항 20.

제 1 항에 있어서, 주어진 입력과 연관되어 있고 상기 중복 모 노드 모두에 존재하는 하나이상의 코드는, 상기 코드가 두 개의 중복 모 노드에 상이한 시퀀스로 발생할 때라도, 그 코드가 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정할 때, 일치하는 것으로서 정의되는 것을 특징으로 하는 시스템.

청구항 21.

제 1 항에 있어서, 트리의 두 개의 모 노드는, 주어진 입력과 연관된 모든 코드가 상기 중복 모 노드 모두에 존재할 때만 중복하는 것으로 정의되고, 그 코드가 동일 시퀀스에서 발생하고, 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정한다는 점에서, 그리고 자 노드가 두 개의 중복 모 노드 각각에 링크된 모든 입력에 대하여, 상기 자 노드가 동일 귀납 센스에서 또 한 중복한다는 점에서 일치하는 것을 특징으로 하는 시스템.

청구항 22.

제 21 항에 있어서, 주어진 입력과 연관되고 상기 중복 모 노드 모두에 존재하는 하나이상의 코드는, 상기 코드가 두 개의 중복 모 노드에서 상이한 시퀀스로 발생할 때라도, 그 코드가 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정할 때, 동일한 것으로서 정의되는 것을 특징으로 하는 시스템.

청구항 23.

제 1 항에 있어서, 대응 모 노드와 연관된 객체를 수정함으로써 자 노드와 연관된 객체를 구성하는데 사용되는 하나이상의 코드는 자 노드와 연관된 구성 객체와 연관된 객체 타입의 명세를 더 포함하고, 두 개의 코드는 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정할 때 그리고, 두 개의 코드 중 최대 하나의 코드는 객체 타입을 지정할 때 일치되는 것으로서 정의되며, 생략된 중복 모 노드내에만 존재하는 자 노드로의 링크 및 임의의 코드에 의해 증대되는 남은 중복 모 노드에 존재하는 코드는 상기 객체 타입의 명세를 포함하는 것을 특징으로 하는 시스템.

청구항 24.

제 23 항에 있어서, 지정되는 객체 타입은 노드를 이용하는 다른 입력 시퀀스로부터 구성 객체에 대응하는 입력 시퀀스를 구별하는 정보를 포함하는 것을 특징으로 하는 시스템.

청구항 25.

사용자에 의해 입력된 모호한 입력 시퀀스를 명확하게 하고,

- (a) 복수의 문자와 각각 관련된 복수의 입력을 가진 사용자 입력 디바이스;

- (b) 입력 시퀀스와 각각 관련된 복수의 객체를 포함하는 메모리;
- (c) 사용자에게 시스템 출력을 묘사하는 디스플레이; 및
- (d) 상기 사용자 입력 디바이스, 메모리, 및 디스플레이에 연결된 프로세서;를 포함하며,

입력 시퀀스는 사용자 입력 디바이스를 조작함으로써 입력이 선택되는 각각의 시간에 생성되고, 생성된 입력 시퀀스는 선택되었던 입력 시퀀스에 대응하며, 상기 생성된 입력 시퀀스는 각각의 입력과 관련된 복수의 문자로 인해 모호한 원문 해석을 가지며,

메모리내의 복수의 객체 각각은 사용 빈도수에 관련되어 있으며, 상기 프로세서는:

- (i) 최대 사용 빈도수를 가지며 각각의 생성된 입력 시퀀스와 관련된 적어도 하나의 객체를 메모리내의 복수의 객체로부터 식별하는 식별 구성요소;
- (ii) 상기 생성된 입력 시퀀스의 원문 해석으로서 각각의 생성된 입력 시퀀스와 관련된 상기 식별 객체 중 적어도 하나를 디스플레이하는 출력 구성요소; 및
- (iii) 모호하지 않은 선택 입력에 사용자 입력 디바이스의 한번이상의 조작을 검출하는 즉시 텍스트 엔트리 디스플레이 위치로의 엔트리에 대하여 상기 식별 객체 중 하나를 선택하는 선택 구성요소;를 포함하고, 각각의 입력 시퀀스와 관련된 객체는 감소하는 사용 빈도수의 순서로 디스플레이되는 명확화 시스템에 있어서,

복수의 입력 중 적어도 하나는 복수의 구두점 문자와 관련된 모호한 구두점 입력이고, 프로세서는, 모호한 구두점 입력의 선택을 검출하는 즉시, 그리고 입력 시퀀스에 첨부하는 즉시, 관련 구두점 문자 중 하나에 종결하는 입력 시퀀스의 관련 원문 해석을 임시로 출력하고, 다음 입력의 선택을 검출하는 즉시, 상기 다음 입력이 지정된 입력의 제 1 세트의 부재일 때, 상기 이전 출력 구두점 문자를 확인하고, 상기 다음 입력이 지정된 입력의 적어도 하나의 대체 세트 중 하나의 부재일 때, 모호한 구두점 입력과 관련된 복수의 구두점 문자 중 다른 문자로 상기 이전 출력 구두점 문자를 대체하고, 여기서 지정된 입력의 상기 제 1 세트 각각은 상기 임시 출력 구두점 문자와 다른 다음 문자의 출력에 관련되어 있고, 지정된 입력의 적어도 하나의 대체 세트의 각각은 상기 임시 출력 구두점 문자와 다른 다음 문자의 출력에 관련되어 있으며, 대체한 구두점 문자는 상기 다음 문자가 하나의 부재인 지정된 입력의 대체 세트에 의해 결정되는 것을 특징으로 하는 시스템.

청구항 26.

제 25 항에 있어서, 지정된 입력의 상기 제 1 세트와 상기 적어도 하나의 대체 세트의 구성은 첨부한 모호한 구두점 입력을 포함하는 상기 입력 시퀀스와 관련된 하나이상의 객체가 상기 메모리에 존재하는지 여부에 의해 결정되는 것을 특징으로 하는 시스템.

청구항 27.

제 25 항에 있어서, 지정된 입력의 상기 제 1 세트와 상기 적어도 하나의 대체 세트의 구성은 첨부한 모호한 구두점 입력을 포함하고, 상기 다음 입력이 모호한 입력일 때 상기 다음 입력을 첨부하는 상기 입력 시퀀스와 관련된 하나이상의 객체가 상기 메모리에 존재하는지 여부에 의해 결정되는 것을 특징으로 하는 시스템.

청구항 28.

제 25 항에 있어서, 지정된 입력의 상기 제 1 세트와 상기 적어도 하나의 대체 세트의 구성은 모호한 구두점 입력의 선택으로 시작하고, 상기 다음 입력이 모호한 입력일 때 상기 다음 입력을 첨부하는 상기 입력 시퀀스와 관련된 하나이상의 객체가 상기 메모리에 존재하는지 여부에 의해 결정되는 것을 특징으로 하는 시스템.

청구항 29.

제 25 항에 있어서, 모호한 구두점 입력은 마침표, 하이픈, 및 어포스트로피와 관련되어 있는 것을 특징으로 하는 시스템.

청구항 30.

제 26 항에 있어서, 프로세서는, 모호한 구두점 입력의 선택을 검출하고 완료 워드에 대응하는 메모리내의 임의의 객체와 관련되지 않은 시퀀스를 형성하기 위해 현재 생성된 입력 시퀀스에 상기 모호한 구두점 입력을 첨부하는 즉시, 상기 관련 구두점 문자 중 하나를 임시로 출력하고,

프로세서는, 문자 's'와 관련되어 있지 않고 하나이상의 알파벳 문자와 관련된 모호한 입력인 다음 입력의 선택을 검출하는 즉시, 이미 출력된 구두점 문자를 하이픈으로 변경하고, 모호한 구두점 입력의 선택에 앞서 현재 생성된 입력 시퀀스의 임시 출력 원문 해석을 확인하고, 상기 다음 모호한 입력을 포함하는 새로운 생성된 입력 시퀀스를 시작하며,

프로세서는, 문자 's'를 포함한 하나이상의 알파벳 문자와 관련된 모호한 입력인 다음 입력의 선택을 검출하는 즉시, 모호한 구두점 입력의 선택에 앞서 현재 생성된 입력 시퀀스의 임시 출력 원문 해석을 확인하고, 상기 다음 모호한 입력을 포함하는 새로운 생성된 입력 시퀀스를 시작하고, 이미 출력된 구두점 문자를 어포스트로피로 대체하고, 상기 새로운 생성 입력 시퀀스의 원문 해석으로서 문자 's'를 임시로 첨부하며, 상기 프로세서는, 하나이상의 알파벳 문자와 관련된 모호한 입력인 다른 다음 입력의 선택을 검출하는 즉시, 이미 출력된 어포스트로피를 하이픈으로 대체하고, 또 다른 다음 모호한 입력에 앞서는 문자 's'를 포함한 하나이상의 알파벳 문자와 관련된 상기 모호한 입력을 포함하는 새로운 생성된 입력 시퀀스를 시작하고, 상기 하이픈에 후속하는 상기 새롭게 시작된 입력 시퀀스와 관련된 메모리내의 객체를 출력하며,

프로세서는, 하나이상의 알파벳 문자와 관련되지 않은 다음 입력의 선택을 검출하는 즉시, 이미 출력된 구두점 문자를 확인하는 것을 특징으로 하는 시스템.

청구항 31.

제 28 항에 있어서, 프로세서는, 모호한 구두점 입력의 선택을 검출하고 메모리내의 임의의 객체와 관련되지 않은 시퀀스를 형성하기 위해 현재 생성된 입력 시퀀스에 상기 모호한 구두점 입력을 첨부하는 즉시, 모호한 구두점 입력의 선택을 검출하기에 앞서 상기 생성된 입력 시퀀스와 관련된 최대 사용 빈도수를 가진 상기 임시로 디스플레이된 객체를 수용하고, 상기 모호한 구두점 입력과 관련된 디폴트 구두점 문자를 임시 출력하고, 모호한 구두점 입력을 포함하는 길이가 긴 새로운 생성된 입력 시퀀스를 설정하는 것을 특징으로 하는 시스템.

청구항 32.

제 25 항에 있어서, 프로세서는, 생성된 입력 시퀀스에서의 모호한 구두점 입력의 발생을 검출하는 즉시, 한번이상 모호하지 않은 선택 입력으로 사용자 입력 디바이스의 조작에 의해 선택될 수 있는 대체 원문 해석의 시퀀스를 생성하고, 대체 원문 해석의 상기 시퀀스의 제 1 시퀀스는, 모호한 구두점 입력과 관련된 구두점 문자 중 하나를 모호한 구두점 입력에 앞서 입력된 생성된 입력 시퀀스로 입력 시퀀스와 관련된 최대 사용 빈도수를 갖는 객체에 첨부하고, 후속하여, 모호한 구두점 입력에 후속하여 입력된 생성된 입력 시퀀스로 입력 시퀀스와 관련된 최대 사용 빈도수를 갖는 객체를 첨부하고, 후속하여, 만들어진 객체가 두 개의 객체의 조합된 빈도수 랭킹의 하향 순서로 만들어지도록, 모호한 구두점 입력에 앞서 입력된 생성된 입력 시퀀스로 입력 시퀀스와 관련된 점차로 더 낮은 사용 빈도수를 갖는 객체 및 모호한 구두점 입력에 후속하여 입력된 생성된 입력 시퀀스로 입력 시퀀스와 관련된 점차로 더 낮은 사용 빈도수를 갖는 객체를 포함하는 객체의 다른 조합에 의하여, 생성되는 것을 특징으로 하는 시스템.

청구항 33.

제 25 항에 있어서, 프로세서는, 모호하지 않은 선택 입력의 제 1 선택에 앞서 하나이상의 모호한 입력의 시퀀스 선택을 검출하는 즉시, 최대 사용 빈도수를 가진 객체를 선택하고, 상기 선택된 텍스트 객체와 단일 후속 스페이스를 디스플레이에 출력하고, 모호하지 않은 선택 입력의 각각의 연속적으로 반복되는 연속 선택을 검출하는 즉시, 다음으로 낮은 사용 빈도수를 가진 객체를 상기 식별된 객체로부터 선택하고, 이미 출력된 단일 스페이스에 앞선 그 디스플레이 위치내의 이미 출력된 텍스트 객체를 각각의 상기 연속적으로 선택된 텍스트 객체로 대체하는 것을 특징으로 하는 시스템.

청구항 34.

제 26 항에 있어서, 프로세서는, 모호한 구두점 입력의 선택을 검출하고 상기 모호한 구두점 입력과 관련된 구두점 문자로 끝나는 최대 사용 빈도수를 가진 메모리내의 완료 워드 객체와 관련된 시퀀스를 형성하기 위해 현재 생성된 입력 시퀀스에 상기 모호한 구두점 입력을 첨부하는 즉시, 그리고, 하나이상의 알파벳 문자와 관련된 모호한 입력인 다음 입력의 선택을 검출하는 즉시, 상기 모호한 구두점 입력의 선택을 포함한 현재 생성된 입력 시퀀스의 원문 해석으로서 상기 완료 워드 객체의 출력을 확인하고, 상기 다음 모호한 입력을 포함하는 새로운 생성된 입력 시퀀스를 시작하는 것을 특징으로 하는 시스템.

청구항 35.

모호한 입력 시퀀스를 명확하게 하고,

- (a) 복수의 문자와 각각 관련된 복수의 입력으로부터 입력 시퀀스를 생성하는 단계;
- (b) 입력 시퀀스와 상대적인 사용 빈도수와 각각 관련된 복수의 객체를 구성하는데 사용되는 데이터를 저장하는 단계;
- (c) 최대 사용 빈도수를 가진 적어도 하나의 후보 객체를 각각의 생성 입력 시퀀스와 관련된 하나이상의 객체로부터 식별하는 단계; 및
- (d) 상기 생성 입력 시퀀스의 원문 해석으로서 각각의 생성 입력 시퀀스와 관련된 적어도 하나의 식별된 후보 객체를 디스플레이하는 단계;를 포함하며,

상기 생성 입력 시퀀스는 각각의 입력과 관련된 복수의 문자로 인해 모호한 원문 해석을 가지며,

모든 객체는 복수의 노드로 구성된 트리 구조에 저장되어 있고, 각각의 노드는 입력 시퀀스와 하나이상의 객체와 관련되어 있고, 복수의 노드는 복수의 경로에 의해 연결되고, 복수의 경로 각각은 베이스 입력 시퀀스와 관련된 모 노드를 추가 입력 및 모 노드의 베이스 입력 시퀀스와 관련된 자 노드에 링크하고, 자 노드와 관련된 객체는 자 노드가 링크된 대응 모 노드와 관련된 객체를 근거로 하고, 자 노드와 관련된 객체는 대응 모 노드에 관련된 객체를 수정하기 위해 메모리에 사전 저장된 코드를 이용하여 구성되고, 대응 모 노드에 관련된 객체를 수정함으로써 자 노드에 관련된 객체를 구성하는데 사용되는 코드는 대응 모 노드에 관련된 객체의 숫자 색인 명세와, 모 노드를 자 노드에 링크시키는 추가 입력에 관련된 문자중 하나의 문자의 숫자 색인 명세를 포함하고,

상기 후보 객체는 적어도 하나의 워드 객체가 상기 생성 입력 시퀀스와 관련되어 있을 때의 워드 객체이고, 상기 후보 객체는 상기 생성 입력 시퀀스와 관련된 워드 객체가 없을 때의 워드 스템 객체인 방법에 있어서,

상기 모든 객체를 트리 구조로 저장하는 상기 단계는 상기 두 개의 모 노드 모두에 존재하는 주어진 입력에 관련된 모든 코드가 그 코드가 동일 시퀀스로 발생하고 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정한다는 점에서 일치할 때, 트리의 두 개의 모 노드가 중복인 중복 모 노드의 제거를 포함하고, 자 노드가 두 개의 모 노드 각각에 링크된 모든 입력에 대하여, 상기 자 노드가 동일 귀납 센스에서 또한 중복이고, 중복 모 노드중 하나의 노드는 메모리내의 트리 구조로부터 제거되고, 생략된 모 노드가 링크로서 재 기록되는 자 노드인 임의의 모 노드로부터 남은 중복 모 노드까지 모두 링크한 상태에서, 남은 중복 모 노드는 생략된 중복 모 노드에만 존재하였던 자 노드로의 링크 및 임의의 코드에 의해 증대되는 것을 특징으로 하는 방법,

청구항 36.

제 35 항에 있어서, 상기 생성 입력 시퀀스의 원문 해석으로서 디스플레이되는 최대 사용 빈도수를 가진 적어도 하나의 식별 후보 객체는 상기 워드 스템 객체의 사용 빈도수가 상기 생성 입력 시퀀스와 관련된 임의의 워드 객체와 관련된 최대 사용 빈도수를 소정 비율만큼 초과할 때, 그리고 상기 생성 입력 시퀀스와 관련된 임의의 워드 객체와 관련된 상기 최대 사용 빈도수가 소정 한계치를 초과하지 않는 곳에서의 워드 스템 객체인 것을 특징으로 하는 방법.

청구항 37.

제 35 항에 있어서, 상기 생성 입력 시퀀스의 원문 해석으로서 임시로 디스플레이되는 최대 사용 빈도수를 가진 적어도 하나의 객체는 하나이상의 문자의 모호하지 않은 생성을 야기하는 임의의 입력이 입력되는 즉시 상기 생성 입력 시퀀스의 원문 해석으로서 출력에 대해 확인되고, 상기 하나이상의 모호하지 않은 문자는 상기 확인된 원문 해석의 출력에 바로 후속하여 출력되는 것을 특징으로 하는 방법.

청구항 38.

제 35 항에 있어서, 대응 모 노드와 관련된 객체를 수정함으로써 자 노드와 관련된 객체를 구성하는데 사용되는 코드는, 그 코드가 자 노드와 관련된 객체를 생성하는 코드 시퀀스의 최종 코드인지에 대한 명세를 더 포함하는 것을 특징으로 하는 방법.

청구항 39.

제 35 항에 있어서, 모 노드에 연결된 자 노드에 대응하는 추가 입력의 일치성과 수는 상기 자 노드의 수와 일치성을 표시하는 유효 키 비트의 필드에 의해 모 노드에 표시되는 것을 특징으로 하는 방법.

청구항 40.

제 39 항에 있어서, 자 노드와 관련된 객체를 생성하는데 사용되는 하나이상의 코드 세트 각각에 상기 자 노드로의 포인터가 바로 후속하고, 하나이상의 코드의 하나이상의 세트와 후속하는 포인터는 상기 자 노드의 수와 일치성을 표시하는 유효 키 비트와 동일 순서로 모 노드내의 메모리에 순차적으로 배치되는 것을 특징으로 하는 방법.

청구항 41.

제 35 항에 있어서, 자 노드와 관련된 객체를 생성하는 코드 시퀀스는, 객체가 상기 객체의 사용 빈도수에 따라 분류되는 시퀀스로 생성되도록 정렬되어 있는 것을 특징으로 하는 방법.

청구항 42.

제 35 항에 있어서, 각각의 입력과 관련된 문자의 색인은 메모리내의 워드 객체에서의 문자의 발생 빈도수의 하향 순서로 문자에 순차적으로 할당되어 있는 것을 특징으로 하는 방법.

청구항 43.

제 35 항에 있어서, 각각의 입력과 관련된 문자의 색인은 복수의 문자를 포함하는 스트링에 할당된 색인을 포함하는 것을 특징으로 하는 방법.

청구항 44.

제 35 항에 있어서, 대응 모 노드와 관련된 객체를 수정함으로써 자 노드와 관련된 객체를 구성하는데 사용되는 코드는 자 노드와 관련된 상기 구성 객체와 관련된 객체 타입의 명세를 더 포함하는 것을 특징으로 하는 방법.

청구항 45.

제 44 항에 있어서, 지정되는 객체 타입은 메모리내의 객체 중 상기 구성 객체를 유일하게 식별하는 코드를 포함하는 것을 특징으로 하는 방법.

청구항 46.

제 44 항에 있어서, 지정되는 객체 타입은 상기 구성 객체의 사용 빈도수에 관한 정보를 포함하는 것을 특징으로 하는 방법.

청구항 47.

제 44 항에 있어서, 지정되는 객체 타입은 상기 구성 객체가 완료 워드인지 여부에 관한 정보를 포함하는 것을 특징으로 하는 방법.

청구항 48.

제 35 항에 있어서, 각각의 입력과 관련된 문자 색인은, 상기 색인 문자가 자 노드와 관련된 객체를 형성하기 위해 첨부되는 대응 모 노드와 관련된 객체내의 바로 앞선 문자에 후속하는 문자의 발생 빈도수의 하향 순서로 문자에 순서적으로 할당되어 있는 것을 특징으로 하는 방법.

청구항 49.

제 35 항에 있어서, 주어진 입력과 관련되어 있고, 상기 중복 모 노드 모두에 존재하는 하나이상의 코드는, 상기 코드가 두 개의 중복 모 노드에서 상이한 시퀀스로 발생할 때라도, 그 코드가 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정할 때, 동일한 것으로서 정의되는 것을 특징으로 하는 방법.

청구항 50.

제 35 항에 있어서, 트리의 두 개의 모 노드는 주어진 입력에 관련된 모든 코드가 상기 중복 모 노드 모두에 존재할 때만 중복으로서 정의되고, 그 코드가 동일 시퀀스로 발생하고, 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정한다는 점에서, 그리고 자 노드가 두 개의 중복 모 노드 각각에 링크된 모든 입력에 대하여, 상기 자 노드는 동일 귀납 센스에서 또한 중복 한다는 점에서 동일한 것을 특징으로 하는 방법.

청구항 51.

제 50 항에 있어서, 주어진 입력과 관련되고, 상기 중복 모 노드 모두에 존재하는 하나이상의 코드는, 상기 코드가 두 개의 중복 모 노드에서 상이한 시퀀스로 발생할 때라도, 그 코드가 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정할 때, 일치하는 것으로서 정의되는 것을 특징으로 하는 방법.

청구항 52.

제 35 항에 있어서, 대응 모 노드와 관련된 객체를 수정함으로써 자 노드와 관련된 객체를 구성하는데 사용되는 하나이상의 코드는 자 노드와 관련된 상기 구성 객체와 관련된 객체 타입의 명세를 더 포함하고, 두 개의 코드는, 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정할 때, 그리고 두 개의 코드 중 최대 하나가 객체 타입을 지정할 때, 동일한 것으로서 정의되고, 생략된 중복 모 노드에만 존재하였던 자 노드로의 링크 및 임의의 코드에 의해 증대되는 남은 중복 모 노드에 존재하는 코드는 상기 객체 타입의 명세를 포함하는 것을 특징으로 하는 방법.

청구항 53.

제 52 항에 있어서, 지정되는 객체 타입은 상기 구성 객체에 대응하는 입력 시퀀스를 노드를 이용하는 다른 입력 시퀀스로부터 구별하는 정보를 포함하는 것을 특징으로 하는 방법.

청구항 54.

모호한 입력 시퀀스를 명확하게 하기 위해 컴퓨터 실행가능 구성요소를 구비하며, 컴퓨터 실행가능 구성요소는

- (a) 복수의 입력을 처리하는 사용자 입력 구성요소;
- (b) 상대적인 사용 빈도수와 입력 시퀀스와 각각 관련된 복수의 객체를 구성하는데 사용되는 데이터를 포함하는 저장 구성요소;
- (c) 최대 사용 빈도수를 가진 적어도 하나의 후보 객체를 각각의 생성 입력 시퀀스와 관련된 하나이상의 객체로부터 식별하는 식별 구성요소; 및
- (d) 상기 생성 입력 시퀀스의 원문 해석으로서 각각의 생성 입력 시퀀스와 관련된 적어도 하나의 식별 후보 객체를 디스플레이하는 출력 구성요소;를 포함하며,

복수의 입력 평균의 각각은 복수의 문자와 관련되어 있고, 상기 생성 입력 시퀀스는 각각의 입력과 관련된 복수의 문자로 인해 모호한 원문 해석을 가지며,

모든 객체는, 복수의 경로에 의해 연결된 복수의 노드로 구성된 트리 구조로 저장되어 있고, 각각의 노드는 입력 시퀀스 및 하나이상의 객체와 관련되어 있고, 복수의 경로 각각은 베이스 입력 시퀀스와 관련된 모 노드를, 추가 입력과 모 노드의 베이스 입력 시퀀스에 관련된 자 노드에 링크시키고, 자 노드와 관련된 객체는 자 노드가 링크된 대응 모 노드에 연관된 객체를 근거로 하고, 자 노드에 연관된 객체는 대응 모 노드에 연관된 객체를 수정하기 위해 메모리에 사전 저장된 코드를 이용하여 구성되고, 대응 모 노드에 연관된 객체를 수정함으로써 자 노드에 연관된 객체를 구성하는데 사용되는 코드는 대응 모 노드에 연관된 객체의 숫자 색인 명세와, 모 노드를 자 노드에 링크시키는 추가 입력에 연관된 문자중 하나의 문자의 숫자 색인 명세를 포함하는 컴퓨터 판독가능 매체에 있어서,

상기 트리 구조는 트리의 두 개의 모 노드가, 두개의 모 노드 모두에 존재하는 주어진 입력에 연관된 모든 코드가 그 코드가 동일 시퀀스로 발생하고, 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정한다는 점에서 동일할 때 중복성인 곳에서, 중복 모 노드를 제거하도록 배열되고, 추가로, 자 노드가 두 개의 모 노드 각각에 링크된 모든 입력에 대하여, 상기 자 노드는 동일 귀납 센스에서 또한 중복이고, 상기 중복 모 노드 중 하나는 메모리내의 트리 구조로부터 생략되고, 남은 중복 모

노드는 생략된 중복 모 노드에만 존재하였던 자 노드로의 링크 및 임의의 코드에 의해 증대되고, 생략된 모 노드가 링크로서 재기록되는 자 노드인 임의의 모 노드로부터 남은 중복 모 노드로 모두 링크하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 55.

제 54 항에 있어서, 상기 생성 입력 시퀀스의 원문 해석으로서 디스플레이되는 최대 사용 빈도수를 가진 적어도 하나의식별 후보 객체는, 상기 워드 스템 객체의 사용 빈도수가 상기 생성 입력 시퀀스와 관련된 임의의 워드 객체와 관련된 최대 사용 빈도수를 소정 비율만큼 초과할 때, 그리고 상기 생성 입력 시퀀스와 관련된 임의의 워드 객체와 관련된 상기 최대 사용 빈도수가 소정 한계치를 초과하지 않는 곳에서 워드 스템 객체인 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 56.

제 54 항에 있어서, 상기 생성 입력 시퀀스의 원문 해석으로서 임시 디스플레이되는 최대 사용 빈도수를 가진 적어도 하나의 객체는 하나이상의 문자의 모호하지 않은 생성을 야기하는 임의의 입력을 입력하는 즉시 상기 생성 입력 시퀀스의 원문 해석으로서 출력에 대하여 확인되고, 상기 하나이상의 모호하지 않은 문자는 상기 확인된 원문 해석의 출력에 바로 후속하여 출력되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 57.

제 54 항에 있어서, 대응 모 노드와 관련된 객체를 수정함으로써 자 노드와 관련된 객체를 구성하는데 사용되는 코드는, 그 코드가 자 노드와 관련된 객체를 생성하는 코드 시퀀스의 최종 코드인지 여부에 관한 명세를 더 포함하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 58.

제 54 항에 있어서, 모 노드에 연결된 자 노드에 대응하는 추가 입력의 수와 일치성은 상기 자 노드의 수와 일치성을 표시하는 유효 키 비트의 필드에 의해 모 노드에 표시되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 59.

제 58 항에 있어서, 자 노드와 관련된 객체를 생성하는데 사용되는 하나 이상의 코드의 각각의 세트에 상기 자 노드로의 포인터가 바로 후속하고, 하나이상의 코드의 하나이상의 세트와 후속하는 포인터는 상기 자 노드의 수와 일치성을 표시하는 유효 키 비트와 동일 순서로 모 노드내에의 메모리에 순차적으로 정렬되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 60.

제 54 항에 있어서, 자 노드와 관련된 객체를 생성하는 코드 시퀀스는, 객체가 상기 객체의 사용 빈도수에 따라 분류되는 순서로 생성되도록 메모리에 정렬되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 61.

제 54 항에 있어서, 각각의 입력과 관련된 문자 색인은 메모리의 워드 객체의 문자 발생 빈도수의 하향 순서로 문자에 순서적으로 할당되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 62.

제 54 항에 있어서, 각각의 입력과 관련된 문자 색인은 복수의 문자를 포함하는 스트링에 할당된 색인을 포함하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 63.

제 54 항에 있어서, 대응하는 모 노드와 관련된 객체를 수정함으로써 자 노드와 관련된 객체를 구성하는데 사용되는 코드는 자 노드와 관련된 상기 구성 객체와 관련된 객체 타입의 명세를 더 포함하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 64.

제 63 항에 있어서, 지정되는 객체 타입은 메모리내의 객체 중 상기 구성 객체를 유일하게 식별하는 코드를 포함하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 65.

제 63 항에 있어서, 지정되는 객체 타입은 상기 구성 객체의 사용 빈도수에 관한 정보를 포함하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 66.

제 63 항에 있어서, 지정되는 객체 타입은 상기 구성 객체가 완료 위드인지 여부에 관한 정보를 포함하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 67.

제 54 항에 있어서, 각각의 입력과 관련된 문자 색인은, 상기 색인 문자가 자 노드와 관련된 객체를 형성하기 위해 첨부된 대응 모 노드와 관련된 객체의 바로 앞선 문자에 후속하는 문자의 발생 빈도수의 하향 순서로 문자에 순서적으로 할당되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 68.

제 54 항에 있어서, 주어진 입력과 연관되고, 상기 중복 모 노드 모두에 존재하는 하나이상의 코드는, 상기 코드가 두 개의 중복 모 노드에서 상이한 시퀀스로 발생할 때라도, 그 코드가 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정할 때, 일치되는 것으로서 정의되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 69.

제 54 항에 있어서, 트리의 두 개의 모 노드는 주어진 입력에 관련된 모든 코드가, 상기 중복 모 노드 모두에 존재할 때만 중복으로서 정의되고, 그 코드가 동일 시퀀스로 발생하고, 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정한다는 점에서, 추가로, 자 노드가 두 개의 중복 모 노드 각각에 링크된 모든 입력에 대하여, 상기 자 노드가 또한 동일 귀납 센스에서 또한 중복이다는 점에서 일치하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 70.

제 69 항에 있어서, 주어진 입력에 연관되고 상기 중복 모 노드 모두에 존재하는 하나이상의 코드는, 상기 코드가 두 개의 중복 모 노드에서 상이한 시퀀스로 발생할 때라도, 그 코드가 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정할 때, 동일한 것으로서 정의되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 71.

제 54 항에 있어서, 대응하는 모 노드와 관련된 객체를 수정함으로써 자 노드와 관련된 객체를 구성하는데 사용되는 하나 이상의 코드는, 자 노드와 관련된 상기 구성 객체와 관련된 객체 타입의 명세를 더 포함하고, 두 개의 코드는, 동일 숫자 객체 색인과 동일 숫자 문자 색인을 지정할 때와, 두 개의 코드 중 최대 하나의 코드가 객체 타입을 지정할 때 일치하는 것으로서 정의되고, 상기 생략된 중복 모 노드에만 존재하였던 자 노드로의 링크 및 임의의 코드에 의해 증대되는 남은 중복 모 노드에 존재하는 코드는 상기 객체 타입의 명세를 포함하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 72.

제 71 항에 있어서, 지정되는 객체 타입은 노드를 이용하는 다른 입력 시퀀스로부터 상기 구성 객체에 대응하는 입력 시퀀스를 구별하는 정보를 포함하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 73.

사용자에 의해 입력된 모호한 입력 시퀀스를 명확하게 하고,

- (a) 복수의 문자와 각각 관련된 복수의 입력으로부터 입력 시퀀스를 생성하는 단계;
- (b) 입력 시퀀스와 사용 빈도수와 관련된 문자 시퀀스를 각각 포함하는 복수의 객체를 저장하는 단계;
- (c) 최대 사용 빈도수를 가지며 각각의 생성 입력 시퀀스와 관련된 적어도 하나의 객체를 복수의 객체로부터 식별하는 단계;
- (d) 상기 생성된 입력 시퀀스의 원문 해석으로서 각각의 생성 입력 시퀀스와 관련된 최대 사용 빈도수를 가진 상기 식별된 객체 중 적어도 하나를 디스플레이하는 단계; 및
- (e) 모호하지 않은 선택 입력의 하나이상의 선택을 검출하는 즉시, 상기 생성 입력 시퀀스의 원문 해석으로서 상기 식별 객체 중 하나를 선택하는 단계;를 포함하며,

상기 생성 입력 시퀀스는 각각의 입력과 관련된 복수의 문자로 인해 모호한 원문 해석을 가지며,

각각의 입력 시퀀스와 관련된 객체는 감소하는 사용 빈도수 순서로 디스플레이되고, 복수의 입력 중 적어도 하나는 복수의 구두점 문자에 연관된 모호한 구두점 입력인 방법에 있어서,

- (a) 모호한 구두점 입력을 포함하는 입력 시퀀스의 발생을 검출하는 단계;
- (b) 상기 모호한 구두점 입력의 선택을 검출하는 즉시, 상기 연관 구두점 문자 중 하나를 임시 출력하는 단계;
- (c) 지정된 입력 중 제 1 세트의 부재인 다음 입력의 선택을 검출하는 즉시, 상기 이전에 출력된 구두점 문자를 확인하는 단계; 및

(d) 지정된 입력의 하나이상의 대체 세트 중 하나의 부재인 다음 입력의 선택을 검출하는 즉시, 상기 이미 출력된 구두점 문자를, 모호한 구두점 입력에 연관된 복수의 구두점 문자 중 다른 구두점 문자로 대체하는 단계;를 포함하며,

지정된 입력의 상기 제 1 세트의 각각은 상기 임시 출력 구두점 문자와 다른 다음 문자의 출력에 연관되고,

지정된 입력의 하나이상의 대체 세트 각각은 상기 임시 출력 구두점 문자와 다른 다음 문자의 출력에 연관되어 있으며, 대체한 구두점 문자는 상기 다음 문자가 하나의 부재인 지정된 입력의 대체 세트에 의해 결정되는 것을 특징으로 하는 방법.

청구항 74.

제 73 항에 있어서, 지정된 입력의 상기 제 1 세트와 상기 하나이상의 대체 세트의 구성은 첨부된 모호한 구두점 입력을 포함한 입력 시퀀스에 연관된 하나이상의 객체가 상기 메모리에 존재하는지 여부에 따라서 결정되는 것을 특징으로 하는 방법.

청구항 75.

제 73 항에 있어서, 지정된 입력의 상기 제 1 세트와 상기 하나이상의 대체 세트의 구성은 상기 다음 입력이 모호한 입력일 때 상기 다음 입력을 더 첨부하고, 상기 첨부된 모호한 구두점 입력을 포함한 입력 시퀀스에 연관된 하나이상의 객체가 상기 메모리에 존재하는지 여부에 따라서 결정되는 것을 특징으로 하는 방법.

청구항 76.

제 73 항에 있어서, 지정된 입력의 상기 제 1 세트와 상기 하나이상의 대체 세트의 구성은 상기 다음 입력이 모호한 입력일 때 상기 다음 입력을 첨부하고, 모호한 구두점 입력의 선택으로 시작하는 입력 시퀀스에 연관된 하나이상의 객체가 상기 메모리에 존재하는지 여부에 따라서 결정되는 것을 특징으로 하는 방법.

청구항 77.

제 74 항에 있어서,

(a) 모호한 구두점 입력의 선택을 검출하는 즉시, 그리고, 완료 워드에 대응하는 메모리내의 임의의 객체와 관련되지 않은 시퀀스를 형성하기 위해 상기 모호한 구두점 입력을 현재 생성된 입력 시퀀스에 첨부하는 즉시, 상기 관련 구두점 문자 중 하나를 임시로 출력하는 단계; 및

(b) 문자 's'와 관련되지 않은 하나이상의 알파벳 문자와 관련된 모호한 입력인 다음 입력의 선택을 검출하는 즉시,

(i) 이미 출력된 구두점 문자를 하이픈으로 변경하고;

(ii) 모호한 구두점 입력의 선택에 앞서 현재 생성된 입력 시퀀스의 임시 출력 원문 해석을 확인하고;

(iii) 상기 다음 모호한 입력을 포함하는 새로운 생성 입력 시퀀스를 시작하는 단계;

(c) 문자 's'를 포함하는 하나이상의 알파벳 문자와 관련된 모호한 입력인 다음 입력의 선택을 검출하는 즉시,

(i) 모호한 구두점 입력의 선택에 앞서 현재 생성된 입력 시퀀스의 임시 출력 원문 해석을 확인하고;

(ii) 상기 다음 모호한 입력을 포함하는 새로운 생성 입력 시퀀스를 시작하고;

(iii) 상기 새로운 생성 입력 시퀀스의 원문 해석으로서 문자 's'를 임시 첨부하고 상기 이전 출력 구두점 문자를 어포스트로 피로 대체하고;

(iv) 하나이상의 알파벳 문자와 연관된 모호한 입력인 또 다른 다음 입력의 선택을 검출하는 즉시, 이미 출력된 어포스트로 피를 하이픈으로 대체하고, 상기 또 다른 다음 모호한 입력에 앞서는 문자 's'를 포함한 하나이상의 알파벳 문자에 연관된 상기 모호한 입력을 포함하는 새로운 생성 입력 시퀀스를 시작하고, 상기 하이픈에 후속하여 상기 새로운 시작 입력 시퀀스에 연관된 메모리내의 객체를 출력하는 단계; 및

(d) 하나이상의 알파벳 문자와 관련되어 있지 않은 다음 입력의 선택을 검출하는 즉시, 상기 이전 출력 구두점 문자를 확인하는 단계;를 더 포함하는 것을 특징으로 하는 방법.

청구항 78.

제 76 항에 있어서,

상기 모호한 구두점 입력의 선택을 검출하는 즉시, 그리고 메모리내의 임의의 객체와 관련되어 있지 않은 시퀀스를 형성하기 위해 현재 생성된 입력 시퀀스에 상기 모호한 구두점 입력을 첨부하는 즉시,

- (a) 상기 모호한 구두점 입력의 선택을 검출하기에 앞서, 상기 생성 입력 시퀀스와 관련된 최대 사용 빈도수를 가진 임시 디스플레이 객체를 수용하는 단계;
- (b) 모호한 구두점 입력과 관련된 디폴트 구두점 문자를 임시 출력하는 단계; 및
- (c) 모호한 구두점 입력을 포함하는 길이가 긴 새로운 생성 입력 시퀀스를 설정하는 단계;를 더 포함하는 것을 특징으로 하는 방법.

청구항 79.

제 73 항에 있어서,

생성된 입력 시퀀스에서의 모호한 구두점 입력의 발생을 검출하는 즉시,

- (a) 한번이상 모호하지 않은 선택 입력에 사용자 입력 디바이스의 조작에 의해 선택될 수 있는 대체 원문 해석을 생성하는 단계; 및
- (b) 한번이상 모호하지 않은 선택 입력에 사용자 입력 디바이스의 조작을 검출하는 즉시, 이러한 대체 원문 해석을 선택하는 단계;를 더 포함하며,

대체 원문 해석의 상기 시퀀스의 제 1 시퀀스는, 모호한 구두점 입력에 앞서 입력된 생성된 입력 시퀀스로 입력 시퀀스와 관련된 최대 사용 빈도수를 갖는 객체에 모호한 구두점 입력과 관련된 구두점 문자 중 하나를 첨부하고, 후속하여, 모호한 구두점 입력에 후속하여 입력된 생성된 입력 시퀀스로 입력 시퀀스와 관련된 최대 사용 빈도수를 갖는 객체를 첨부하고, 후속하여, 만들어진 객체가 두 개의 객체의 조합된 빈도수 랭킹의 하향 순서로 만들어지도록, 모호한 구두점 입력에 앞서 입력된 생성된 입력 시퀀스로 입력 시퀀스와 관련된 점차로 더 낮은 사용 빈도수를 갖는 객체 및 모호한 구두점 입력에 후속하여 입력된 생성된 입력 시퀀스로 입력 시퀀스와 관련된 점차로 더 낮은 사용 빈도수를 갖는 객체를 포함하는 객체의 다른 조합에 의하여, 생성되는 것을 특징으로 하는 방법.

청구항 80.

제 73 항에 있어서,

- (a) 모호한 선택 입력의 제 1 선택에 앞서 하나이상의 모호한 입력의 시퀀스 선택을 검출하는 즉시, 최대 사용 빈도수를 가진 객체를 선택하여, 상기 선택 텍스트 객체와 단일의 후속 스페이스를 디스플레이에 출력하는 단계; 및
- (b) 모호하지 않은 선택 입력의 각각의 연속적으로 반복되는 연속적인 선택을 검출하는 즉시,
 - (i) 상기 식별 객체로부터 다음의 보다 낮은 사용 빈도수를 가진 객체를 선택하고,
 - (ii) 각각의 상기 연속적으로 선택된 텍스트 객체로, 상기 이미 출력된 단일 스페이스에 앞선 디스플레이 위치내의 상기 이미 출력된 텍스트 객체를 대체하는 단계;를 더 포함하는 것을 특징으로 하는 방법.

청구항 81.

제 74 항에 있어서,

모호한 구두점 입력의 선택을 검출하는 즉시 그리고, 상기 모호한 구두점 입력과 관련된 구두점 문자로 끝나는 최대 사용 빈도수를 가진 메모리내의 완료 워드 객체와 관련된 시퀀스를 형성하기 위해 현재 생성된 입력 시퀀스에 상기 모호한 구두점 입력을 첨부하는 즉시, 그리고, 하나이상의 알파벳 문자와 관련된 모호한 입력인 다음 입력의 선택을 검출하는 즉시,

- (a) 모호한 구두점 입력의 선택을 포함한 상기 현재 생성된 입력 시퀀스의 원문 해석으로서 상기 완료 워드 객체의 출력을 확인하는 단계; 및
- (b) 상기 다음 모호한 입력을 포함하는 새로운 생성 입력 시퀀스를 시작하는 단계;를 더 포함하는 것을 특징으로 하는 방법.

청구항 82.

모호한 입력 시퀀스를 명확하게 하기 위해 컴퓨터 실행가능 구성요소를 구비한 컴퓨터 판독가능 매체에 있어서, 컴퓨터 실행가능 구성요소는:

- (a) 복수의 문자와 각각 연관된 복수의 입력을 처리하는 사용자 입력 구성요소;
- (b) 사용 빈도수와 입력 시퀀스와 각각 관련된 복수의 객체를 포함하는 저장 구성요소;
- (c) 최대 사용 빈도수를 가지며 각각의 생성 입력 시퀀스와 관련된 적어도 하나의 객체를 복수의 객체로부터 식별하는 식별 구성요소;
- (d) 상기 생성 입력 시퀀스의 원문 해석으로서 각각의 생성 입력 시퀀스와 관련된 식별 객체 중 적어도 하나를 디스플레이 하는 출력 구성요소; 및
- (e) 모호하지 않은 선택 입력의 하나이상의 선택을 검출하는 즉시, 상기 생성 입력 시퀀스의 원문 해석으로서 상기 식별 객체 중 하나를 선택하는 선택 구성요소;를 포함하며,

상기 생성 입력 시퀀스는 각각의 입력과 관련된 복수의 문자로 인해 모호한 원문 해석을 가지며,

각각의 입력 시퀀스와 관련된 객체는 감소하는 사용빈도수의 순서로 디스플레이되고,

모호한 입력 시퀀스를 명확하게 하는 컴퓨터 실행가능 구성요소는,

- (a) 복수의 입력 중 적어도 하나의 입력은 복수의 구두점 문자에 연관된 모호한 구두점 입력이고,

- (b) 상기 모호한 구두점 입력의 선택을 검출하는 즉시, 관련 구두점 문자 중 하나를 임시로 출력하고,
 - (c) 지정된 입력의 제 1 세트의 부재인 다음 입력의 선택을 검출하는 즉시, 이미 출력된 구두점 문자를 확인하고,
 - (d) 지정된 입력의 하나이상의 대체 세트 중 하나의 세트의 부재인 다음 입력의 선택을 검출하는 즉시, 이미 출력된 구두점 문자를, 모호한 구두점 입력에 연관된 복수의 구두점 문자의 다른 문자로 대체하고,
- 지정된 입력의 상기 제 1 세트의 각각은 상기 임시 출력 구두점 문자와 다른 다음 문자의 출력과 연관되어 있고, 지정된 입력의 상기 하나이상의 대체 세트 각각은 상기 임시 출력 구두점 문자와 다른 다음 문자의 출력에 연관되어 있으며, 대체한 구두점 문자는 상기 다음 입력이 하나의 부재인 지정된 입력의 대체 세트에 의해 결정되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 83.

제 82 항에 있어서, 모호한 입력 시퀀스를 명확하게 하는 컴퓨터 실행가능 구성요소를 구비한 컴퓨터 판독가능 매체로서, 컴퓨터 실행가능 구성 요소는:

지정된 입력의 상기 제 1 세트와 상기 하나이상의 대체 세트의 구성은 첨부한 모호한 구두점 입력을 포함하는 상기 입력 시퀀스에 연관된 하나이상의 객체가 상기 저장 구성 요소에 존재하는지 여부에 의해 결정되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 84.

제 82 항에 있어서, 모호한 입력 시퀀스를 명확하게 하는 컴퓨터 실행가능 구성요소를 구비한 컴퓨터 판독가능 매체로서, 컴퓨터 실행가능 구성 요소는:

지정된 입력의 상기 제 1 세트와 상기 하나이상의 대체 세트의 구성은 첨부한 모호한 구두점 입력을 포함하고, 상기 다음 입력이 모호한 입력일 때 상기 다음 입력을 추가로 첨부하는 상기 입력 시퀀스에 연관된 하나이상의 객체가 상기 저장 구성 요소에 존재하는지 여부에 의해 결정되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 85.

제 82 항에 있어서, 모호한 입력 시퀀스를 명확하게 하는 컴퓨터 실행가능 구성요소를 구비한 컴퓨터 판독가능 매체로서, 컴퓨터 실행가능 구성요소는:

지정된 입력의 상기 제 1 세트와 상기 하나이상의 대체 세트의 구성은 상기 다음 입력이 모호한 입력일 때 상기 다음 입력을 첨부하고, 모호한 구두점 입력의 선택으로 시작하는 상기 입력 시퀀스에 연관된 하나이상의 객체가 상기 저장 구성요소에 존재하는지 여부에 의해 결정되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 86.

제 83 항에 있어서,

- (a) 모호한 구두점 입력의 선택을 검출하는 즉시, 그리고 완료 워드에 대응하는 저장 구성요소내의 임의의 객체와 관련되어 있지 않은 시퀀스를 형성하기 위해 현재 생성된 입력 시퀀스에 상기 모호한 구두점 입력을 첨부하는 즉시, 출력 구성요소는 상기 관련 구두점 문자 중 하나를 임시로 출력하며,

(b) 문자 's'와 관련되어 있지 않은 하나이상의 알파벳 문자와 관련된 모호한 입력인 다음 입력의 선택을 검출하는 즉시, 출력 구성요소는 이미 출력된 구두점 문자를 하이픈으로 변경하고, 모호한 구두점 입력의 선택에 앞서 현재 생성된 입력 시퀀스의 임시 출력 원문 해석을 확인하고, 상기 다음 모호한 입력을 포함하는 새로운 생성 입력 시퀀스를 시작하며,

(c) 문자 's'를 포함하는 하나이상의 알파벳 문자와 관련된 모호한 입력인 다음 입력의 선택을 검출하는 즉시, 출력 구성요소는, 모호한 구두점 입력의 선택에 앞서 현재 생성된 입력 시퀀스의 임시 출력 원문 해석을 확인하고, 상기 다음 모호한 입력을 포함하는 새로운 생성 입력 시퀀스를 시작하고, 상기 이미 출력된 구두점 문자를 어포스트로피로 대체하고, 상기 새로운 생성 입력 시퀀스의 원문 해석으로서 문자 's'를 임시 첨부하며, 입력 구성요소가 하나이상의 알파벳 문자에 연관된 모호한 입력인 또 다른 다음 입력의 선택을 검출하는 즉시, 상기 출력 구성요소는, 상기 이미 출력된 어포스트로피를 하이픈으로 대체하고, 입력 구성요소는 상기 또 다른 다음 모호한 입력에 앞서 문자 's'를 포함한 하나이상의 알파벳 문자에 연관된 상기 모호한 입력을 포함하는 새로운 생성 입력 시퀀스를 시작하고, 출력 구성요소는 상기 하이픈에 후속하여 상기 새롭게 시작된 입력 시퀀스에 연관되어 있는 것으로서 식별 구성 요소에 의해 식별된 객체를 출력하고,

(d) 하나이상의 알파벳 문자와 관련되어 있지 않은 다음 입력의 선택을 검출하는 즉시, 출력 구성요소는 이미 출력된 구두점 문자를 확인하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 87.

제 85 항에 있어서, 모호한 구두점 입력의 선택을 검출하는 즉시, 그리고 저장 구성요소내의 임의의 객체와 관련되어 있지 않은 시퀀스를 형성하기 위해 상기 모호한 구두점 입력을 현재 생성된 입력 시퀀스에 첨부하는 즉시, 출력 구성요소는, 모호한 구두점 입력의 선택을 검출하기에 앞서 상기 생성 입력 시퀀스와 관련된 최대 사용 빈도수를 가진 임시 디스플레이 객체를 수용하고, 상기 모호한 구두점 입력과 관련된 디폴트 구두점 문자를 임시 출력하며, 상기 모호한 구두점 입력을 포함하는 길이가 긴 새로운 생성 입력 시퀀스를 설정하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 88.

제 82 항에 있어서, 상기 생성 입력 시퀀스로 상기 모호한 구두점 입력의 발생을 검출하는 즉시,

(a) 상기 출력 구성요소는 한번이상 모호하지 않은 선택 입력의 선택에 의해 선택될 수 있는 대체 원문 해석 시퀀스를 생성하고,

(b) 상기 선택 구성 요소는 한번이상 모호하지 않은 선택 입력에 사용자 입력 디바이스의 조작을 검출하는 즉시, 이러한 대체 원문 해석 중 하나를 선택하고,

상기 대체 원문 해석의 상기 시퀀스의 제 1 시퀀스는 모호한 구두점 입력에 앞서 입력되었던 상기 생성 입력 시퀀스내의 입력 시퀀스에 연관된 최대 사용 빈도수를 가진 객체에, 모호한 구두점 입력에 연관된 구두점 문자중 하나를 첨부하고, 그 다음, 모호한 구두점 입력에 후속하여 입력된 상기 생성 입력 시퀀스내의 입력 시퀀스에 연관된 최대 사용 빈도수를 가진 객체를 첨부하고, 그 다음, 생성된 객체가 상기 두 개의 객체의 결합된 사용 빈도수 랭킹의 하향 순서로 생성되도록, 모호한 구두점 입력에 앞서 입력되었던 상기 생성 입력 시퀀스내의 입력 시퀀스에 연관된 점진적으로 낮아지는 사용 빈도수를 가진 객체와, 모호한 구두점 입력에 후속하여 입력되었던 상기 생성 입력 시퀀스내의 입력 시퀀스에 연관된 점진적으로 낮아지는 사용 빈도수를 가진 객체를 포함하는 객체들을 결합함으로써, 생성되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 89.

제 82 항에 있어서,

(a) 모호하지 않은 선택 입력의 제 1 입력에 앞서 하나이상의 모호한 입력의 시퀀스의 선택을 검출하는 즉시, 선택 구성요소는 최대 사용 빈도수를 가진 객체를 선택하고, 출력 구성요소는 상기 선택된 텍스트 객체와 단일의 후속 스페이스를 디스플레이에 출력하며;

(b) 모호하지 않은 선택 입력의 각각의 연속적으로 반복되는 연속적인 선택을 검출하는 즉시, 선택 구성요소는 다음으로 사용빈도수가 낮은 객체를 상기 식별 객체로부터 선택하고, 출력 구성요소는 상기 연속적으로 선택된 텍스트 객체로, 상기 이미 출력된 단일 스페이스에 앞선 디스플레이 위치에서의 상기 이미 출력된 텍스트 객체를 대체하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 90.

제 83 항에 있어서, 모호한 구두점 입력의 선택을 검출하는 즉시, 그리고 상기 모호한 구두점 입력과 관련된 구두점 문자로 끝나는 최대 사용 빈도수를 가진 저장 구성요소내의 완료 워드 객체와 관련된 시퀀스를 형성하기 위해 상기 현재 생성된 입력 시퀀스에 모호한 구두점 입력을 첨부하는 즉시, 그리고, 하나이상의 알파벳 문자와 관련된 모호한 입력인 다음 입력의 선택을 검출하는 즉시, 출력 구성요소는 모호한 구두점 입력의 선택을 포함하는 현재 생성된 입력 시퀀스의 원문 해석으로서 상기 완료 워드 객체의 출력을 확인하고, 상기 다음 모호한 입력을 포함하는 새로운 생성 입력 시퀀스를 시작하는 것을 특징으로 하는 컴퓨터 판독가능 매체.

청구항 91.

삭제

청구항 92.

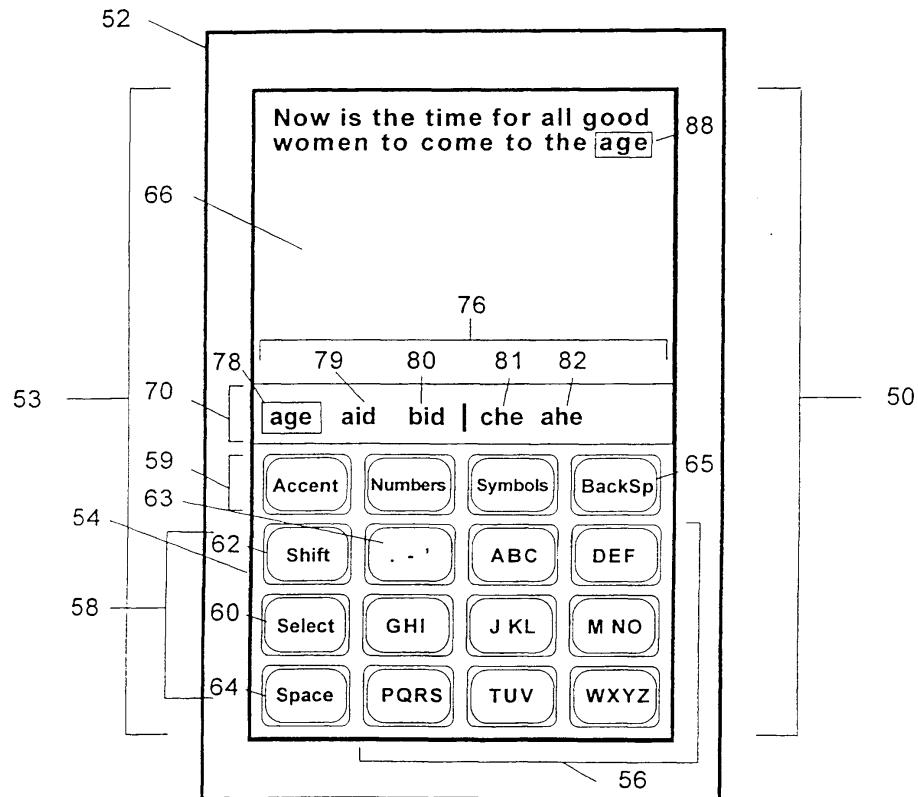
삭제

청구항 93.

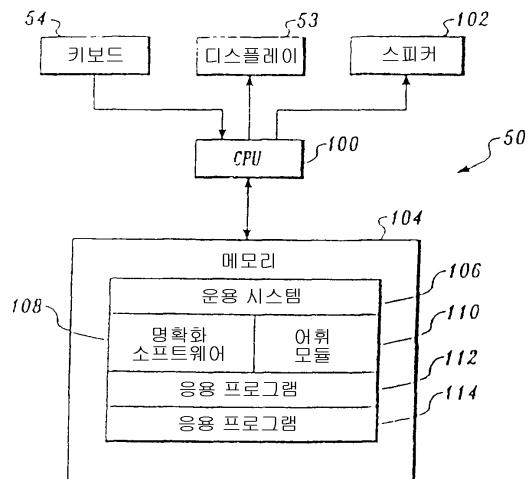
삭제

도면

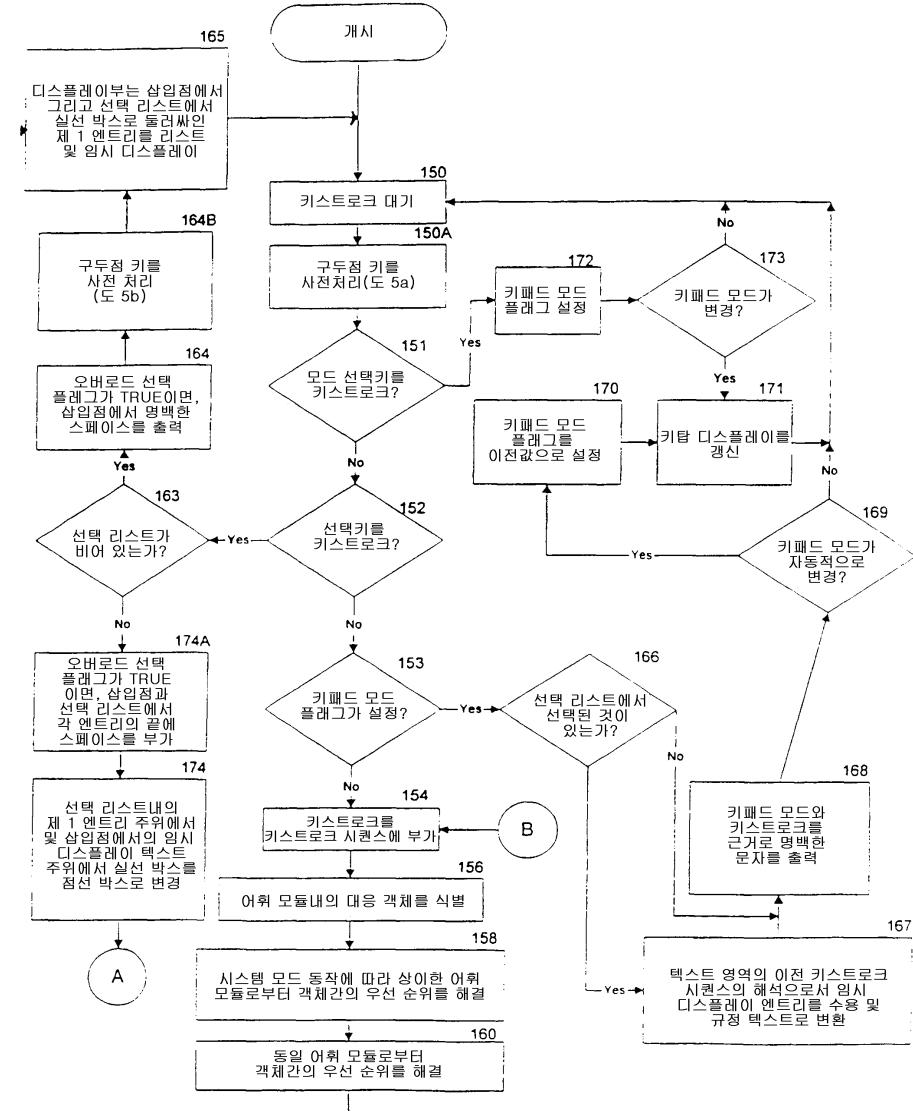
도면1



도면2

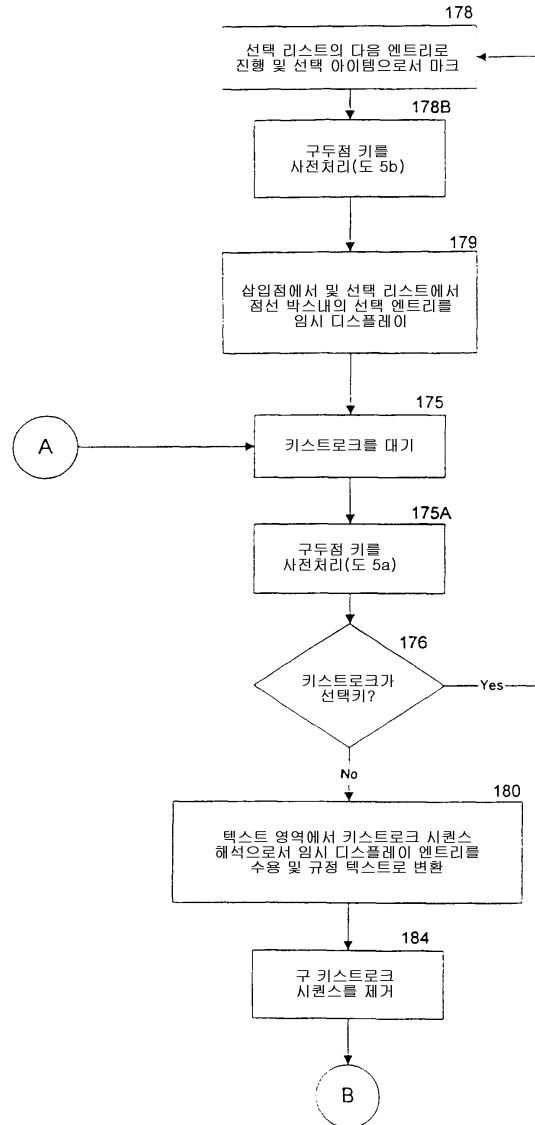


도면3a



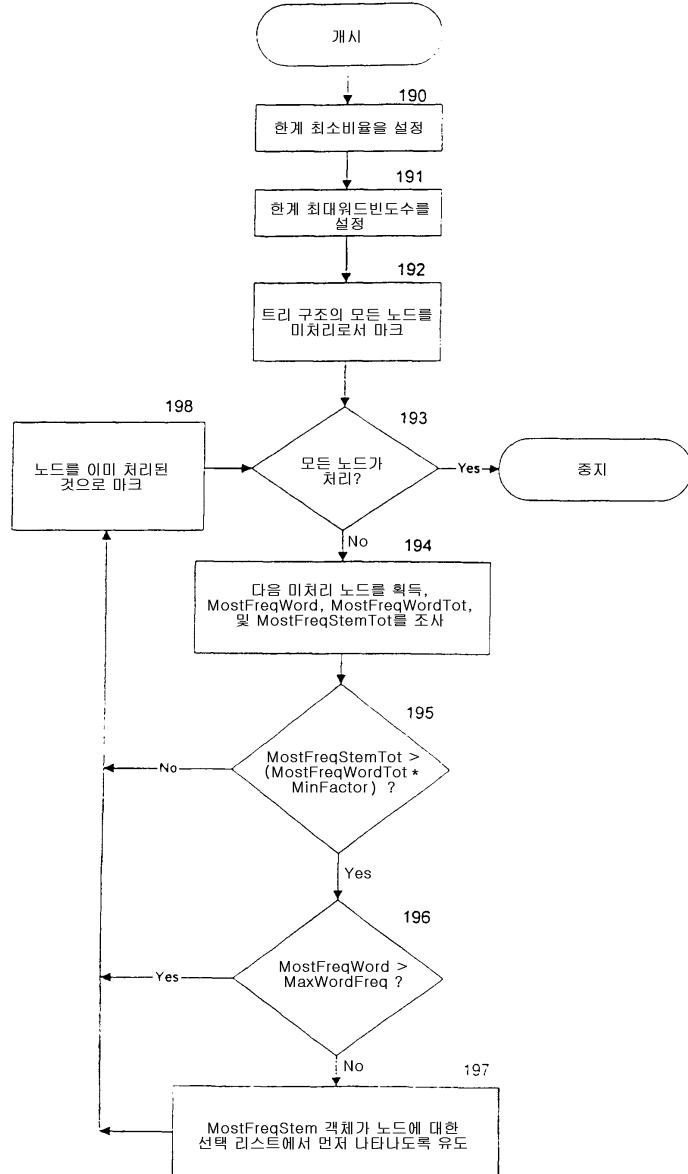
도 3

도면3b

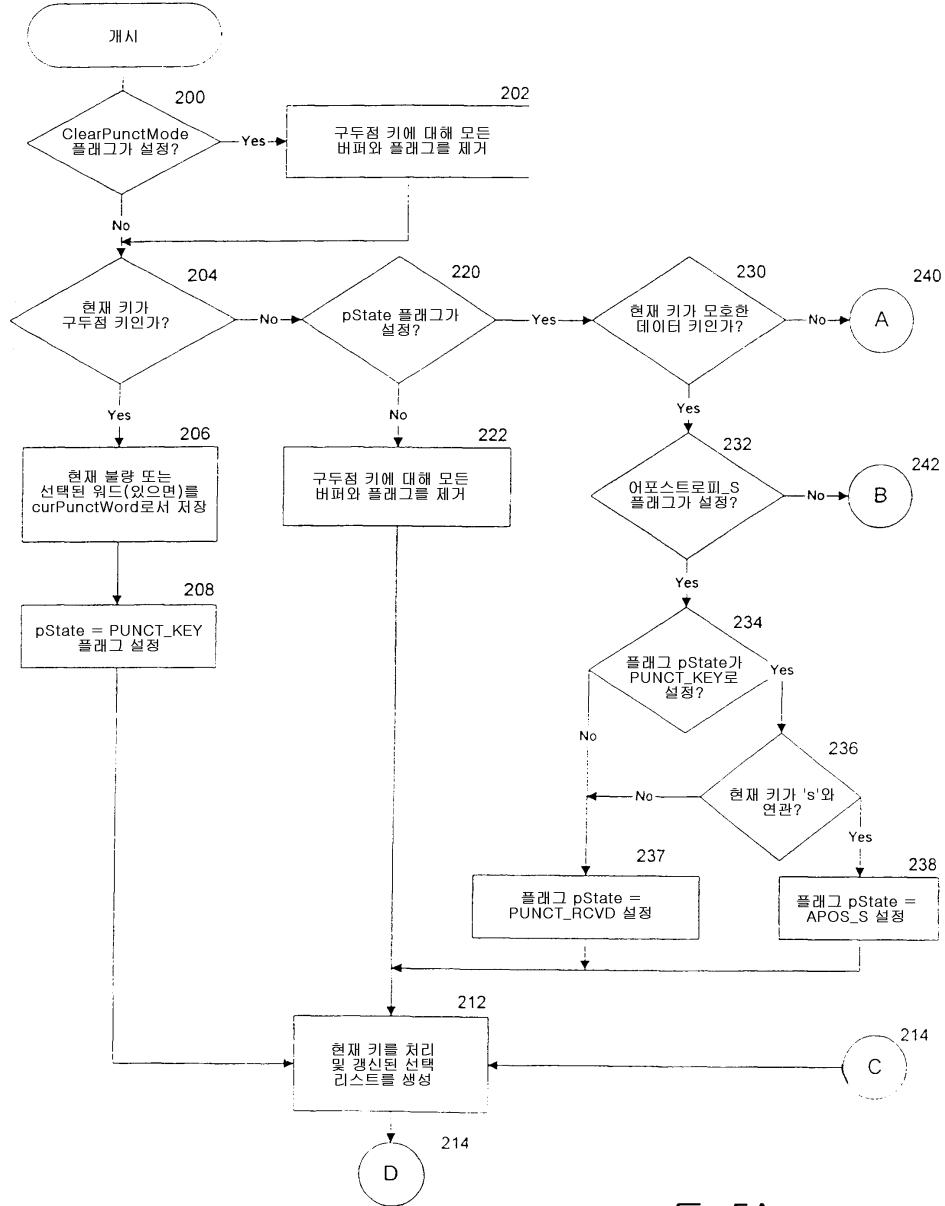


도 3 (계속)

도면4

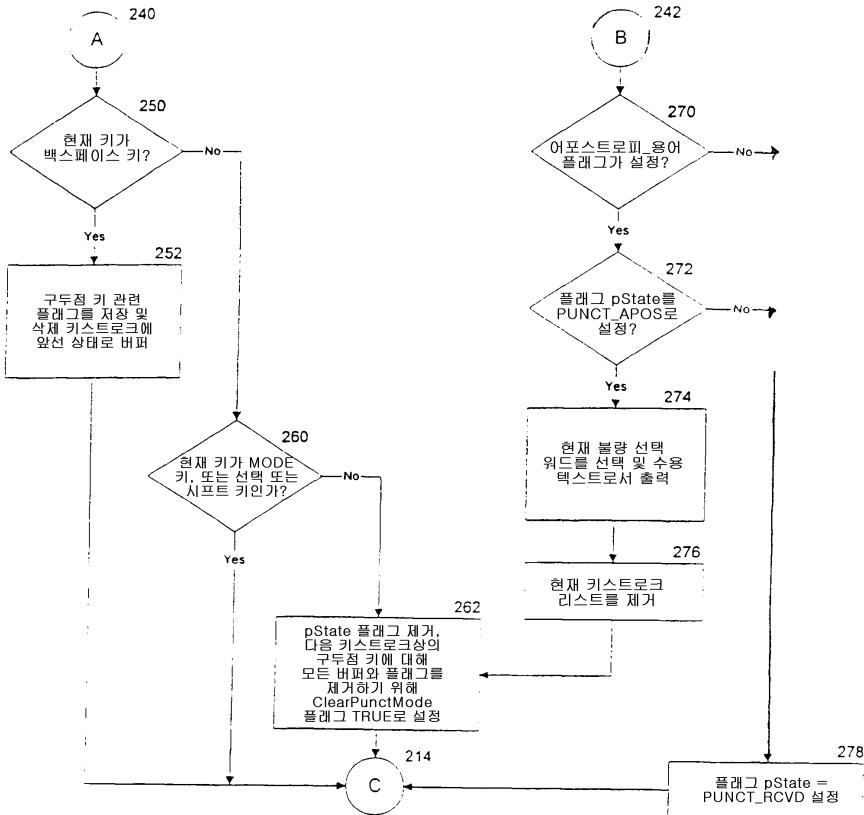


도면5a



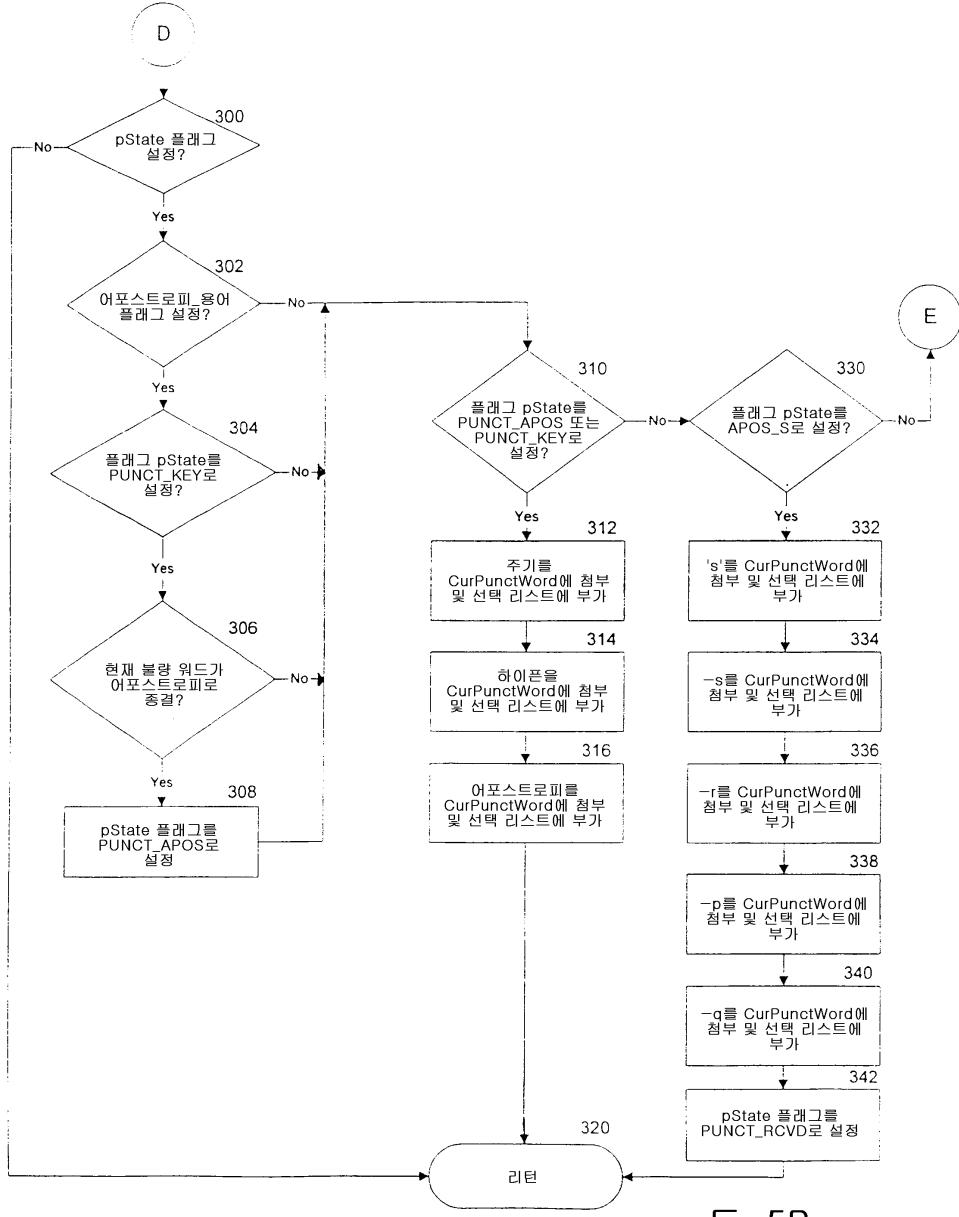
도 5A

도면5b



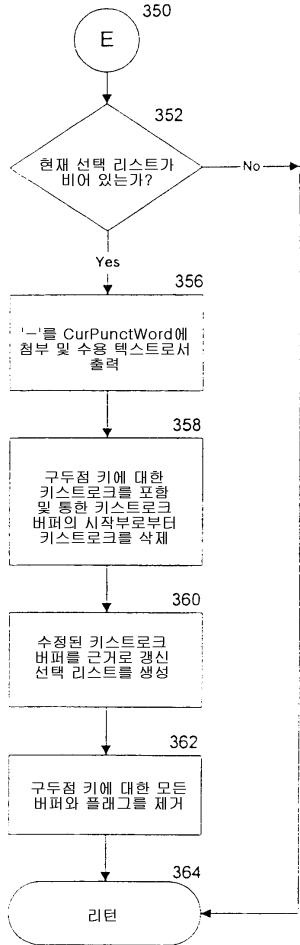
도 5A (계속)

도면5c



도 5B

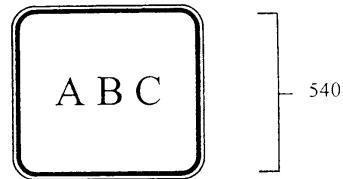
도면5d



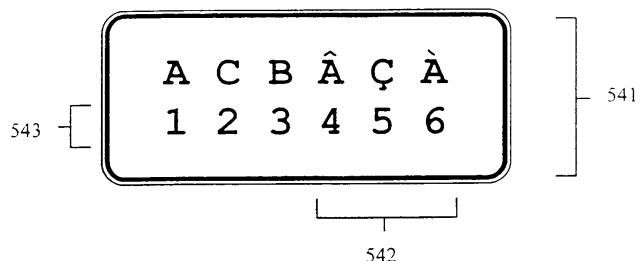
도 5B (계속)

도면6

율리적 키



논리 키
(프랑스어)



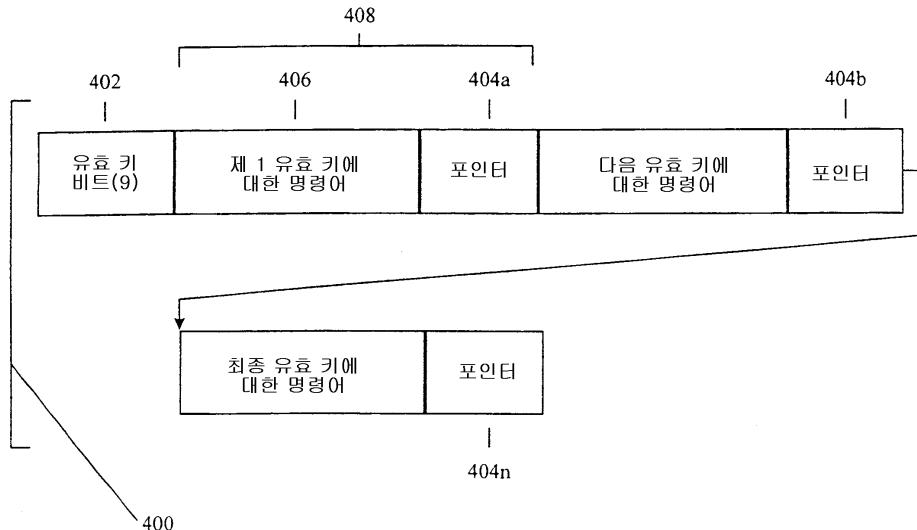
도면7

프랑스어에 대한
논리 기호 색인표

A large vertical bracket on the left side of the grid, with the number '550' written next to it. The grid itself is enclosed in a large horizontal bracket on the right side, with the number '552' written next to it. At the bottom center of the grid is a small horizontal bracket with the number '551' written next to it.

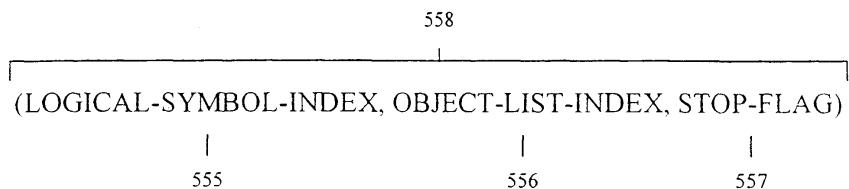
	1	2	3	4	5	6	7
1	.	-	'				
2	a	c	b	â	ç	à	
3	e	é	d	f	è	ê	ë
4	i	g	h	î	ï		
5	l	j	k				
6	n	o	m	ô	œ		
7	r	s	p	q			
8	t	u	v	û	ú	ù	ü
9	x	z	y	w			

도면8a

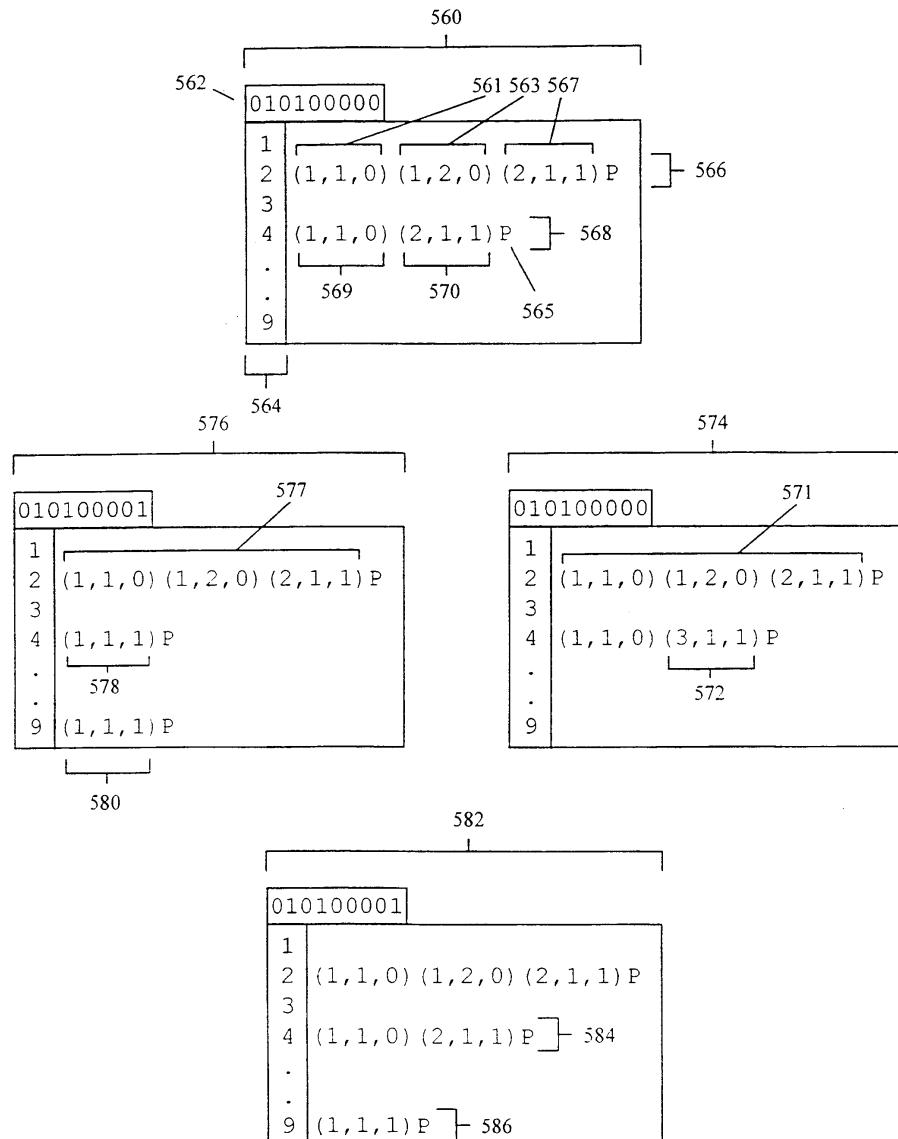


도면8b

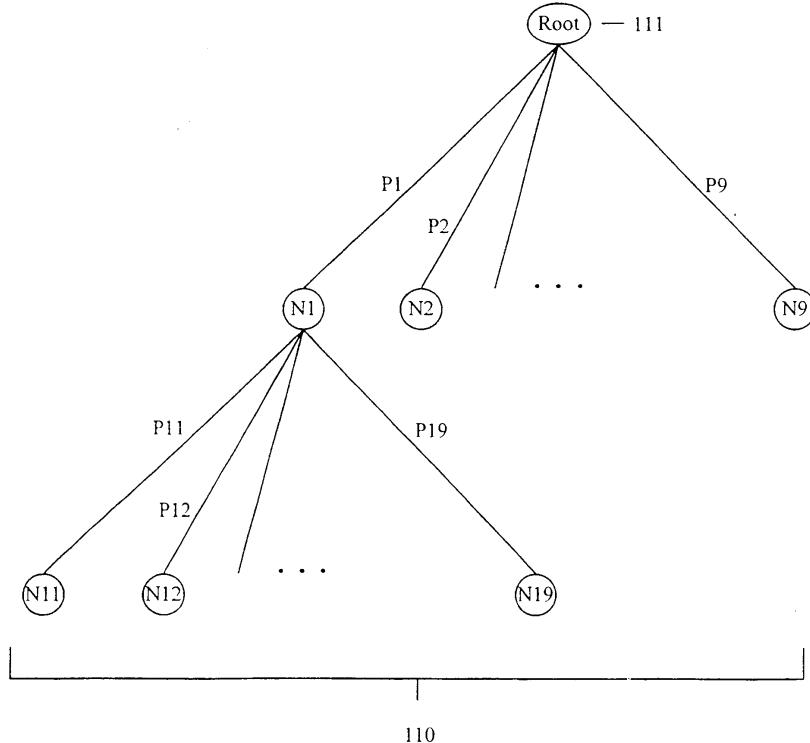
명령어 어의



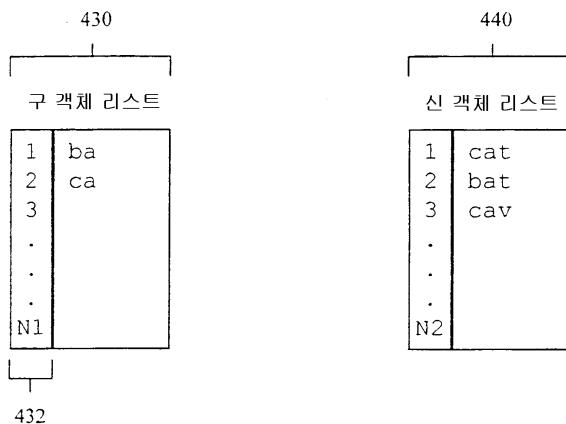
도면9



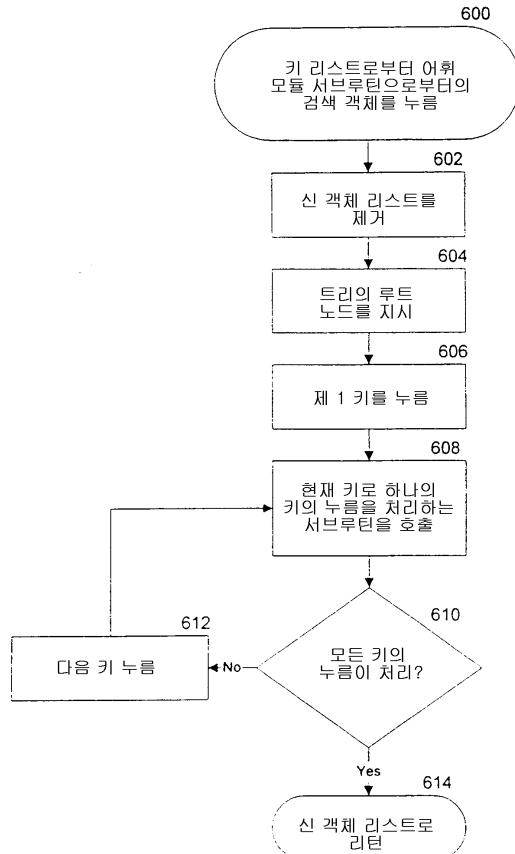
도면10



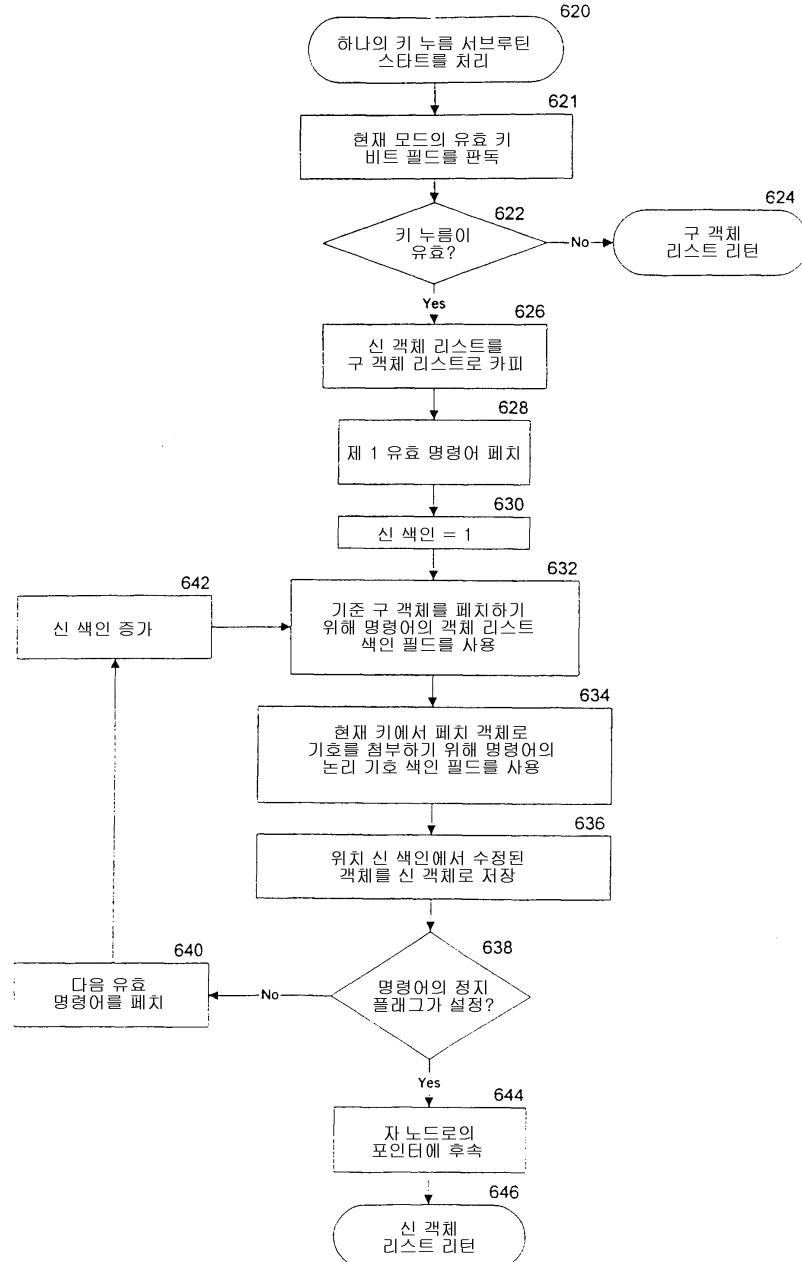
도면11



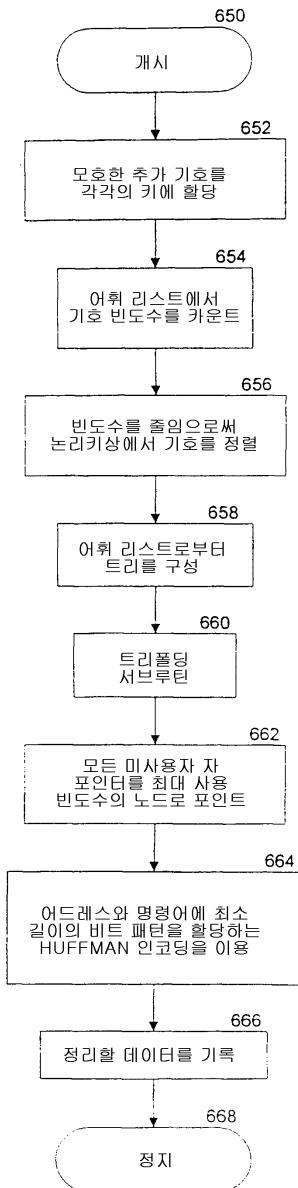
도면12



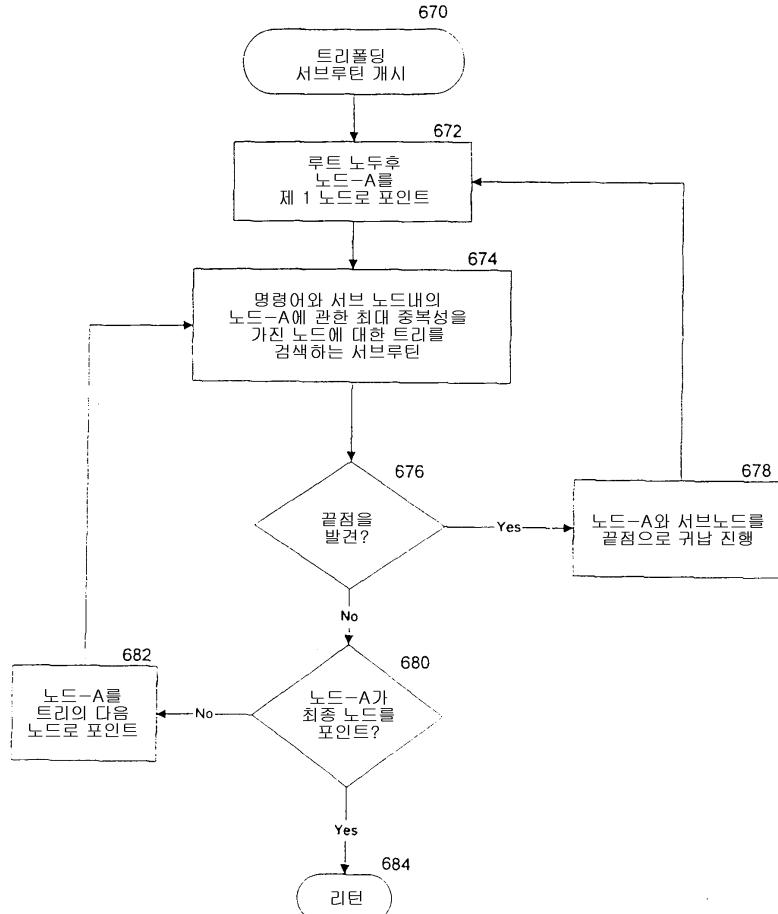
도면13



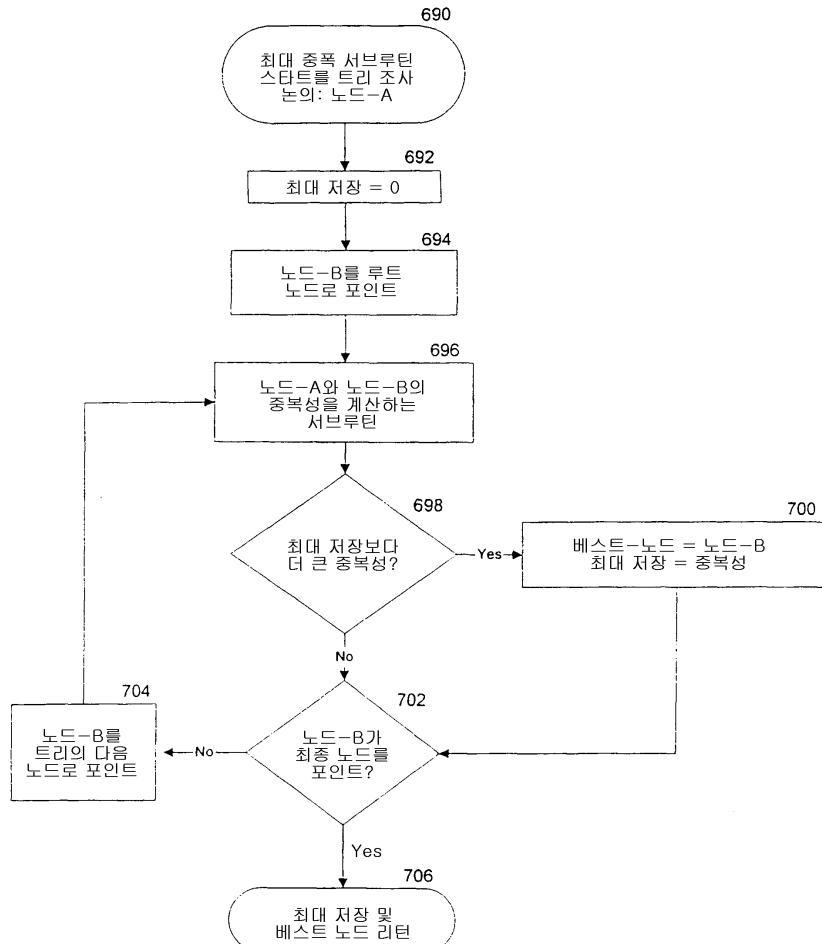
도면14



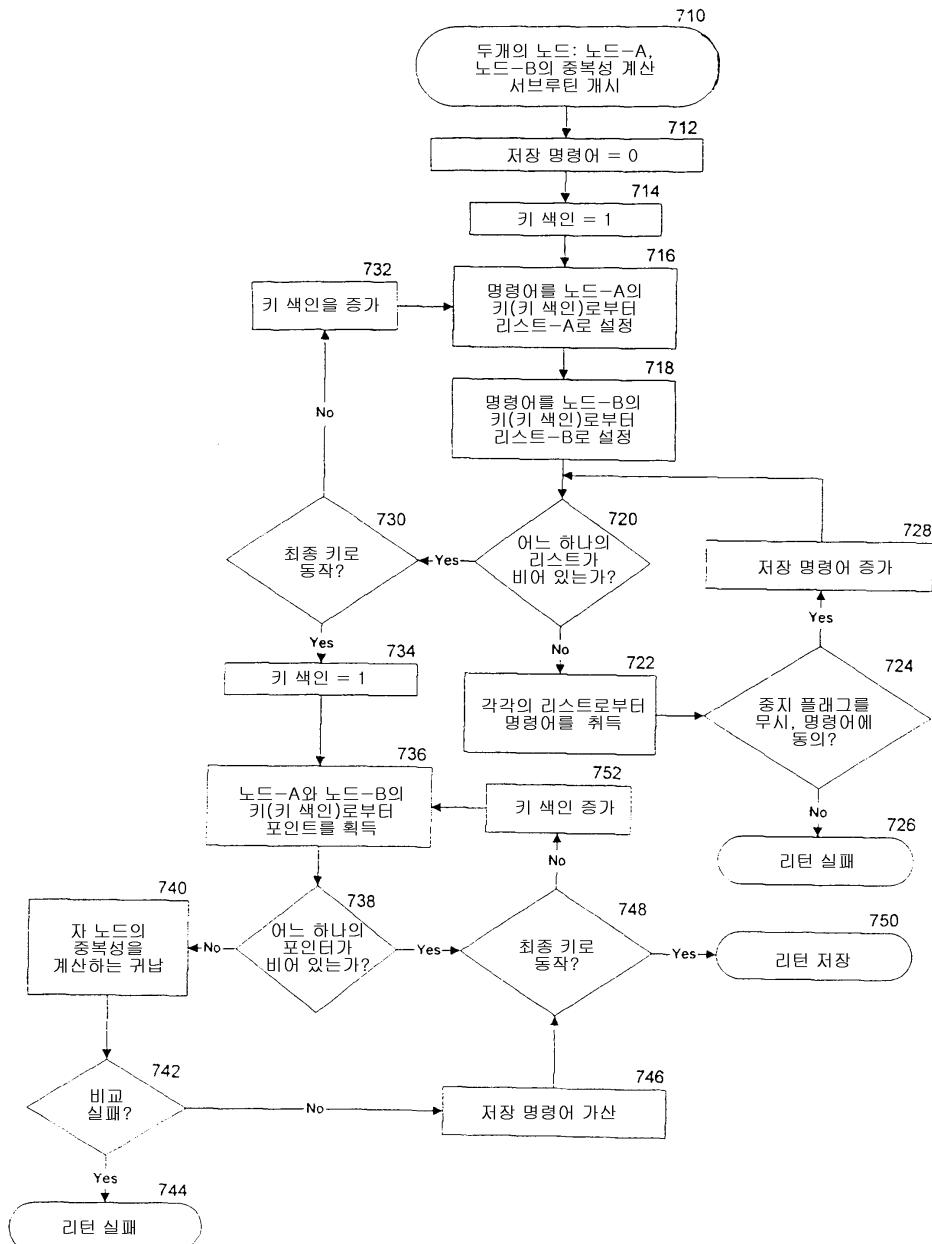
도면15



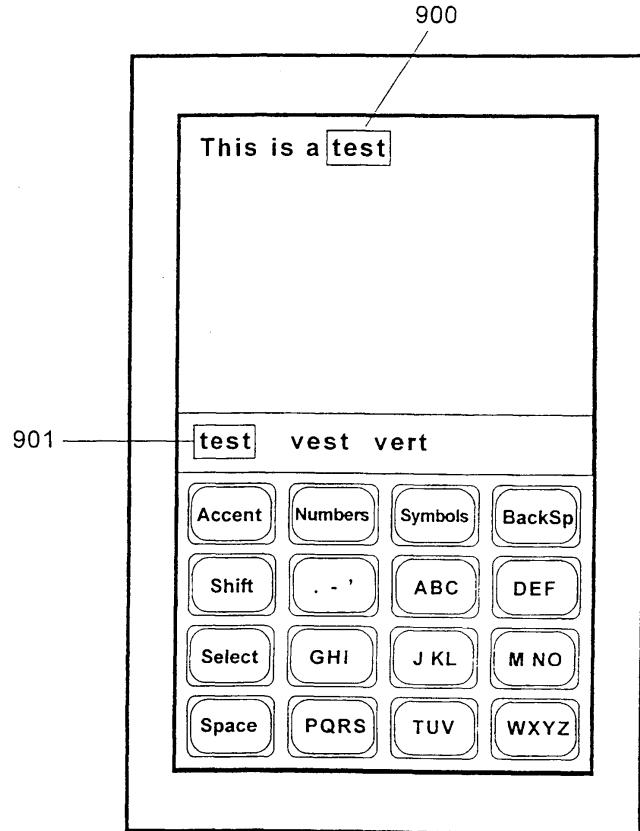
도면16



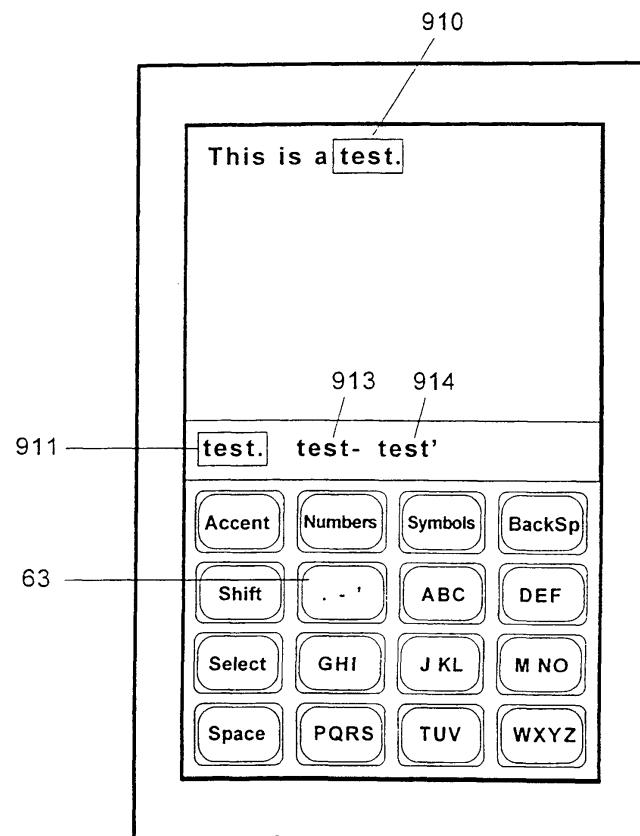
도면17



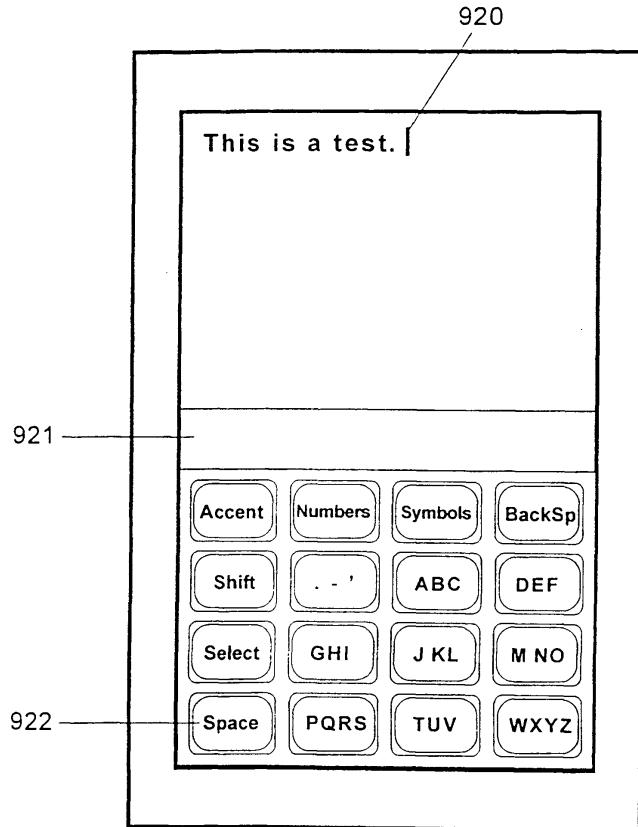
도면18a



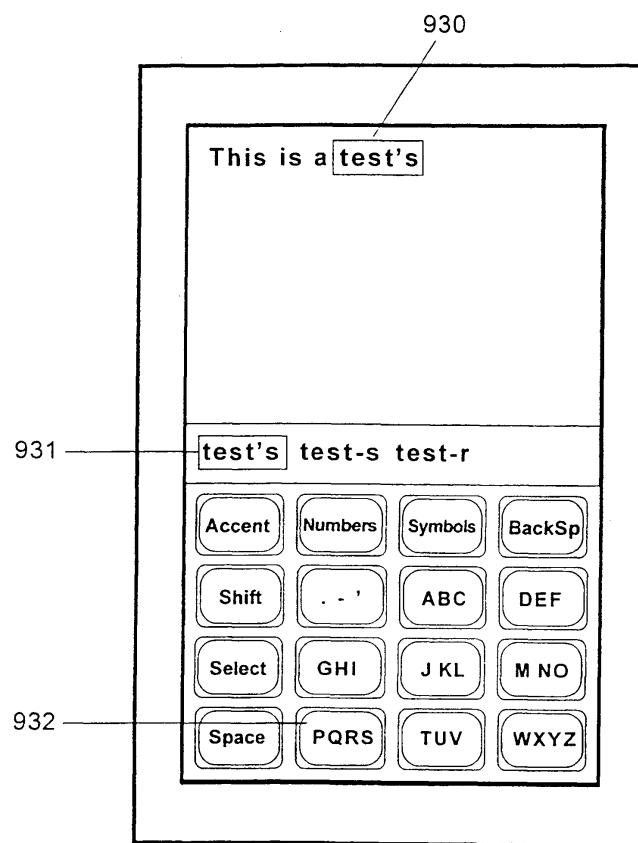
도면18b



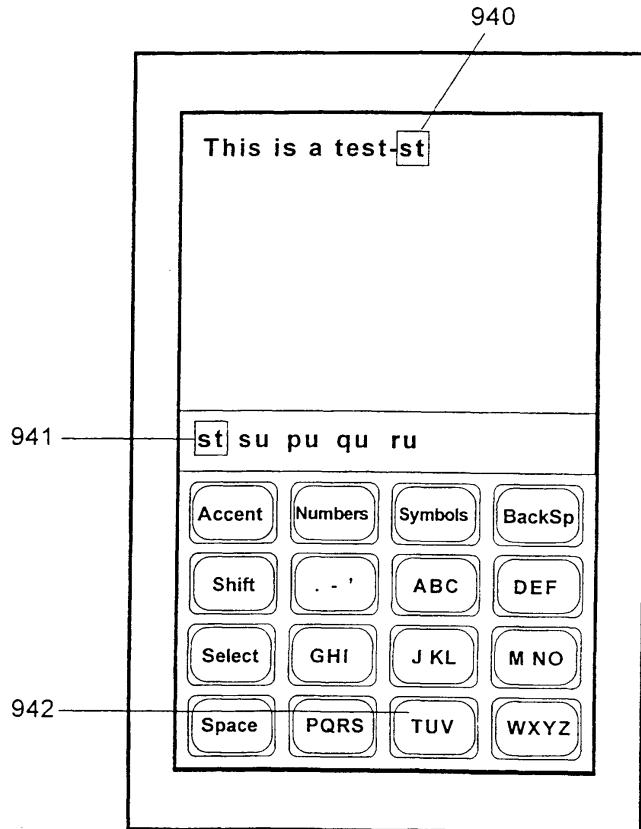
도면18c



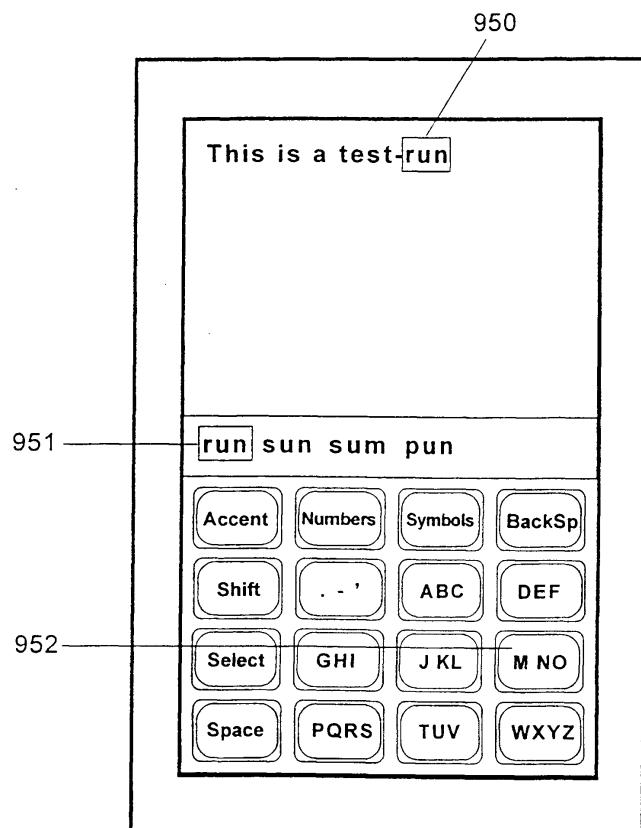
도면18d



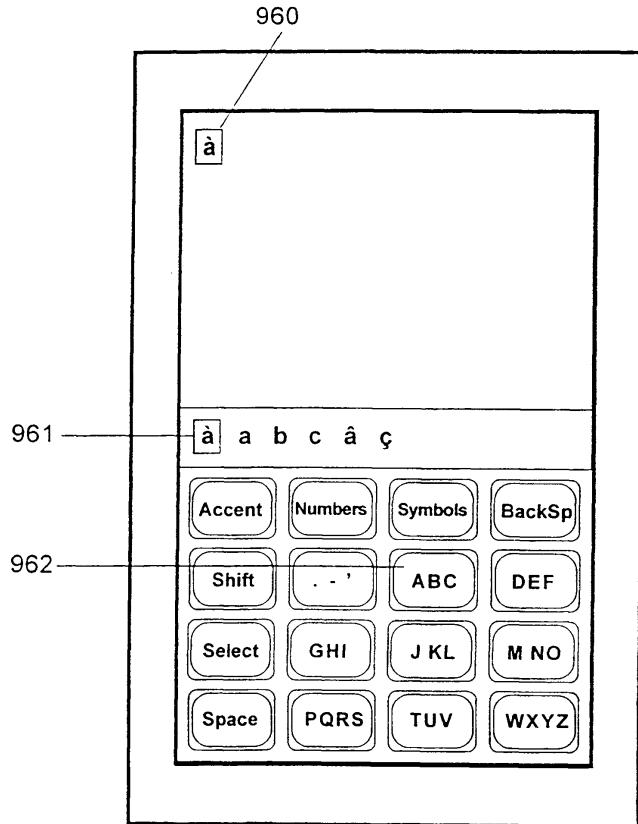
도면18e



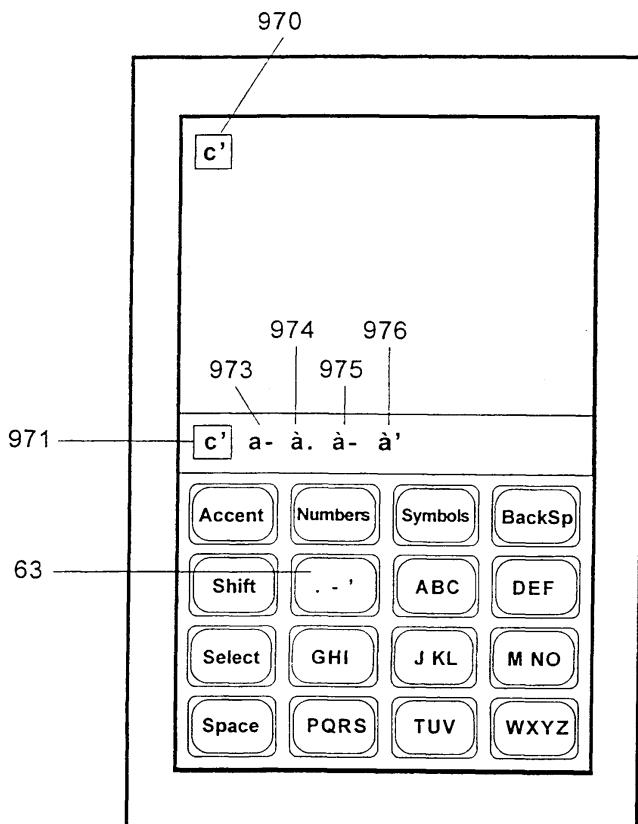
도면18f



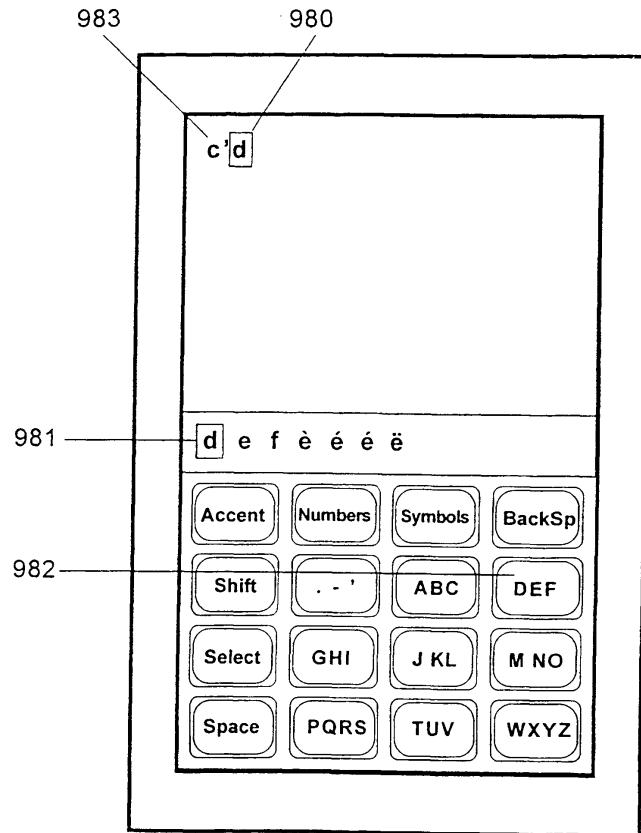
도면18g



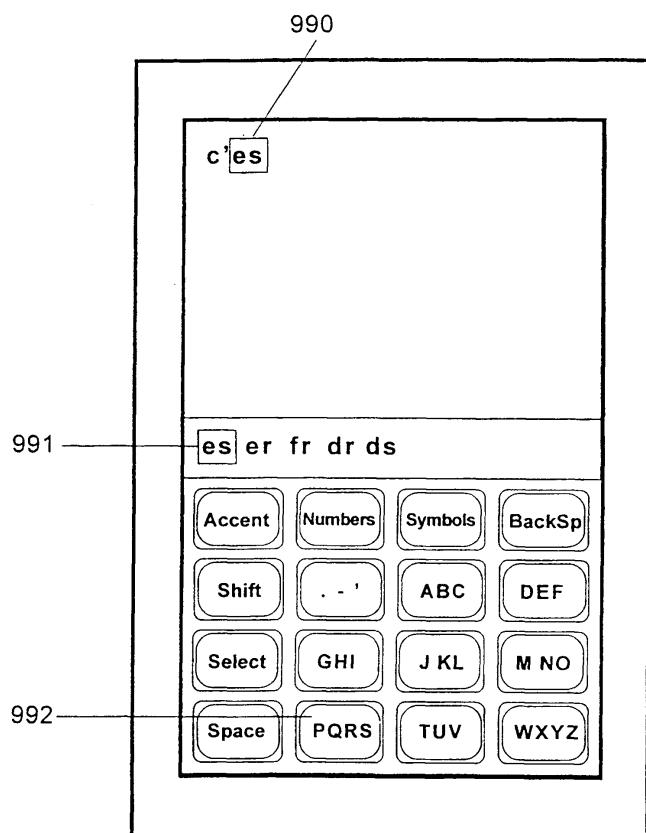
도면18h



도면18i



도면18j



도면18k

