

CHR for Spoken and other Biological Languages

Veronica Dahl

Department of Computer Science
Simon Fraser University
Burnaby, B.C., Canada
`veronica@cs.sfu.ca`

Abstract. CHR is now well established as an invaluable tool for computing and other formal science applications. Much less studied is their use in the humanistic sciences. In this article we bring together our personal view on applying Constraint Reasoning as embodied in CHR to joining the humanistic with the formal sciences, through the link of language— understood in a broad sense as including both spoken languages and molecular biology languages. and we try to distill from these heterogeneous enterprises some common threads that can possibly lead to an embryonic model of humanistic investigation through CHR. The applications we cover include such themes as a dual processing scheme for both human and biological languages; decoding nucleic acid strings through human language; DNA replication as a model for computational linguistics; multi-disciplinary biological knowledge representation for early cancer diagnosis; RNA-inspired analysis of poetry; parsing medical text into de-identified databases; and biological concept-formation.

Keywords: CHR, CHR_G, assumptions, abduction, Hyprolog, nucleic acid mining, computational linguistics, concept formation, parsing, medical text processing,

1 Introduction

Instantaneous communication made possible by contemporary technology is favouring more cross-disciplinary interactions than ever, as well as accelerating those within each field. And it was high time, because it is becoming clearer that some disciplines simply need to join forces with others. For instance, an unprecedented volume growth of biological data over the past few years (most notably, human language text produced in the form of articles, books, web sites, etc., and genetic code text in nucleic acid language, such as DNA sequences) has created formidable challenges for their timely and interrelated processing. Traditional methods in biology for processing and making sense of such information can no longer keep up with the exponentially growing information tsunami. Methodologies that pertain to the Natural Language Understanding field of AI are now being exploited to analyze biological sequences, which is uncovering similarities between the languages of molecular biology and spoken languages such as English. Such similarities might help explain the curious fact first discussed in [24],

that many techniques used in bioinformatics, even if developed independently, may be seen to be grounded in linguistics.

This observation, still valid, resonates deeply with this author, whose obsession with language in all forms has led her, through sometimes unconventional interdisciplinary enterprises that often try to bridge the humanistic and the formal sciences, to search for ever higher while executable levels of description that can transfer into different disciplines.

During this search, the CHR paradigm [19] as embedded in CHR_G [7] and HyProlog [5] has become central in many ways, which we discuss in this paper. The main extensions we use are abduction (the unsound but useful inference of p as a possible explanation for q given that p implies q) and assumptions (resources that are globally available as from their inception while being backtrackable) [17, 5], both subject to consistency with a special type of facts: integrity constraints. These extensions allow us to move beyond the limits of classical logic to explore possible cause and what-if scenarios— as befits the needs for flexibility of linguistic applications.

2 Background

2.1 CHR_G

CHR Grammars, or CHR_Gs for short, are to CHR what Definite Clause Grammars (DCGs) are to Prolog, and are executed as CHR programs that provide robust parsing with an inherent treatment of ambiguity.

For instance, the CHR_G rules

```
token(leucine)::> codon([u,u,a]).
token(leucine)::> codon([u,u,g]).
```

are equivalent to the CHR rules:

```
token(X0,X1,leucine)==> codon(X0,X1,[u,u,a]).
token(X0,X1,leucine)==> codon(X0,X1,[u,u,g]).
```

where the word-boundary arguments, $X0$ and $X1$, are now explicit.

2.2 Hyprolog

Hyprolog [5] is an extension of Prolog and of CHR with assumptions and abduction.

Abduction is agreed upon as a powerful technique in logic programming but its actual use in practice appears to be rather limited since most available systems are research prototypes implemented using inefficient metaprogramming techniques. Assumptive logic programming is related to abduction but provides explicit creation and consumption of hypotheses plus scoping principles inspired by linear logic. It can be hard-wired as in Bin-Prolog [20] or incorporated into any Prolog version with some loss of speed.

As an example of the use of assumptions, here is a graph walking program which avoids loops simply because assumed facts are usable (i.e. consumed upon their successful unification with a goal) at most once. "c" stands for "connected, so $c(\text{Node}, \text{NodeList})$ denotes that Node is connected with each element of NodeList. Assumptions are preceded by a plus sign, and consumptions by a minus sign.

```
path(X,X,[X]).
path(X,Z,[X|Xs]):-!linked(X,Y),path(Y,Z,Xs).

linked(X,Y):- -c(X,Ys), member(Y,Ys).

start(Xs):-
  +c(1,[2,3]), +c(2,[1,4]), +c(3,[1,5]), +c(4,[1,5]),
  path(1,5,Xs).
```

By executing `?-start(Xs)`, we will avoid loops like 1-2-1 and 1-2-4-1 and obtain the expected paths:

```
Xs=[1,2,4,5];
Xs=[1,3,5]
```

In [5] we showed that abduction and assumptions can be integrated with traditional Prolog programs without any significant slow-down in execution speed or other burdens for the programmer.

This was achieved by using a trivial extension for abducible predicates and assumptions written in CHR. From the user's end, the notation and processing remain consistent with ordinary Prolog, including for integrity constraints, which are expressed in CHR (as embedded in Prolog). The only visible difference with ordinary Prolog is that certain predicates are declared as abducibles, that assumptions are identified through notation (plus for assuming and minus for consuming), and that abducibles (as is standard with abducibles) never appear as clause heads.

For instance, to abduce that it rained or that a sprinkler was on from the information that the grass is wet, under an integrity constraint that if the trees are dry it cannot have rained last night, all we need to write (aside from calling the Hyprolog module and declaring our abducibles as such) is:

```
grass(wet):- rained_last_night.
grass(wet):- sprinkler_was_on.

trees(dry), rained_last_night ==> fail.
```

To the best of our knowledge, HyProlog provides one of the most efficient implementations of abductive logic programming, perhaps the most efficient one, and the price for this is a limited support of negation, as detailed in [5].

3 Language Understanding

3.1 On spoken languages and the two cultures

Humans interpret spoken languages through clues much beyond the literal meaning strictly conveyed in the text being interpreted; for instance metaphor, irony, and pronoun reference are usually directly understood even if metaphor often points to a different domain than the discourse's, irony conveys roughly the opposite of what is literally being said, and several entities may potentially be referred to by the same pronoun. Contextual and pragmatic knowledge present in the hearer's mind but hard for a computer to store and pertinently retrieve makes humans quite unbeatable in language understanding compared to computers. Two major breakthroughs have nevertheless considerably advanced the state-of-the-art in the past few decades:

- Logic based processing tools have made it possible for high level formulations of human languages to be directly executable while remaining close to human formulations, such as those familiar to linguists (documented for instance yearly at the European Summer School of Logic, Language and Informatics series).
- The computer industry's bonanza in terms of speed and memory has made brute force approaches more affordable and in many cases successful. In particular, statistical and probabilistic language models have had engineering success in providing an accurate simulation of some linguistic phenomena [21].

These two developments are sometimes seen as competing, and there is an ongoing discussion in linguistic circles as to which is "best", cf. for instance [21].

It is our thesis that applying CHR –or its grammatical counterpart, CHR_G– to processing language promotes combinations of these two approaches which can largely keep the best of both worlds. We shall develop this thesis in what follows.

3.2 Parsing spoken human languages

CHR allows for high level while executable descriptions of language much as logic grammars do. In addition, its bottom-up emphasis probably makes it simpler to grasp for minds with a penchant for the concrete, and for those for whom the word logic equates with "difficult".

As well, it facilitates interaction between different language levels, by making it possible to access several constraints, possibly coming from disparate language modules, through the same rule. Thus, CHR rules lend themselves beautifully to the specialized task of describing linguistic formalisms which view a grammar as a set of constraints, and parsing as a constraint satisfaction problem. One interesting example are Property Grammars (PG) [1], described through properties between constituents plus conditions under which some can be relaxed. Faced

with incomplete or incorrect input, the parser still delivers results rather than failing and indicates the reasons of anomaly through a list of satisfied and a list of violated properties. For instance, a PG parse of the noun phrase “every blue moon” results in a set of satisfied properties (e.g. *linear precedence* holds between the determiner and the noun, between the adjective and the noun, and between the determiner and the adjective; the noun’s requirement for a determiner is satisfied, etc.) and a set of unsatisfied properties, which is empty for this example. Some properties can be relaxed, e.g. a language tutoring system geared to Spanish speaking students might want to relax the linear precedence constraint between the adjective and the noun, so that for instance “Every moon blue” would be accepted even though ungrammatical in English, but the violation would be indicated by placing the violated relationship in the set of unsatisfied properties, thus signaling to the student an unexpected word ordering in the language he or she is trying to learn.

One of the problems with constraint-based approaches is that constraints are usually expressed over high-level objects or structures. This is the case for example in HPSG [22], in which complex feature-structures must first be built before constraints can be evaluated. Similarly, Optimality Theory [23] also generates a set of structures (or candidate structures) and then uses constraints to filter this set. In our approach, any constraint can be evaluated at any time for any set of categories. Such an evaluation dynamically adds new information: the satisfaction of a *selection* constraint instantiates the syntactic category it describes. But this instantiation is conceived almost as a side effect of evaluation: satisfying constraints does not rely on the knowledge of the upper-level category. In other words, the hierarchical information is no longer preponderant in the parsing process. This means that one can evaluate subsets of constraints, for example in the case of applications that only need NP recognition. Achieving robustness in the face of incomplete information or noise is thus also made easier.

We have developed a parsing scheme for PGs in terms of CHR [11] which validates the model of property centered parsing with respect to efficiency, while preserving the level of generality of this theory. To illustrate our parsing scheme for PGs, we now show the core rule, which combines two consecutive categories (one of which is of type XP or obligatory) into a third, by testing each of the properties on the pair and creating the new property lists through property inheritance [11]. Its form is described in Fig. 1.

This rule first tests that one of the two categories is of type XP (a phrase category) or obligatory (i.e., the head of an XP), and that the other category is an allowable constituent for that XP. It then successively tests each of the PG properties among those categories, incrementally building as it goes along the lists of satisfied and unsatisfied properties. Finally, it infers a new category of type XP spanning both these categories, with the finally obtained **Sat** and **Unsat** lists as its characterization.

In practice, this rule unfolds into two symmetric parts, to accommodate the situation in which the XP category appears before the category Cat which is to be incorporated into it.

```

cat(Cat,Features1,Graph1,Sat1,Unsat1):(Start1,End1),
cat(Cat2,Features2,Graph2,Sat2,Unsat2):(End1,End2) :>
  xp_or_obli(Cat2,XP), ok_in(XP,Cat),
  acceptable(precedence(XP,Start1,End1,End2,Cat,Cat2,Sat1,Unsat1,SP,UP,BP)),
  acceptable(dependency(XP,Start1,End1,End2,Cat,Features1,Cat2,
    Features2,SP,UP,SD,UD,BD)),
  build_tree(XP,Graph1,Graph2,Graph,ImmDaughters),
  acceptable(unicity(Start,End2,Cat,XP,ImmDaughters,SD,UD,SU,UU,BU)),
  acceptable(requirement(Start,End2,Cat,XP,ImmDaughters,SU,UU,SR,UR,BR)),
  acceptable(exclusion(Start,End2,Cat,XP,ImmDaughters,SR,UR,Sat,Unsat,BE))
| cat(XP,Features2,Graph,Sat,Unsat).

```

Fig. 1. New Category Inference

We note that direct interpretation, moreover, guarantees a better evolution of the system: it can better adjust to changes in the theory and to experimental stages. This has for instance allowed us to add new semantic properties for specific application purposes [14].

4 From linguistic to medical applications: early diagnoses, de-identification

Not only language levels can be simultaneously accessed by a single rule: also non-linguistic components of an AI system can. This facilitates, for instance, cooperation between knowledge repositories and linguistic components of a same system, e.g. the grammatical component of an AI system can determine the semantic type of a linguistic argument in collaboration with an ontology component of the same system, or even with web search.

Thus, [2] takes inspiration from the linguistic developments above reported (in particular, from the author's rendition of Property Grammars [11], which relies exclusively on constraints, controls the parse through head-driven analysis, and provides a direct interpretation, while preserving all theoretical properties at the implementation level) to propose a CHR based model of multidisciplinary information which can combine heterogeneous clues about the development of oral and lung cancer at the early stages along with the patients medical history and behavioral risk analysis, for a more accurate diagnostic of the probability of the lesions advancing to cancer or not. Each element of data involved in the analysis presents very different characteristics, and the model is robust in that it will still reach useful conclusions even when not all of the data is available for a given patient. Underlying this model is an extension of the CHR-based Concept Formation model [12], which evolved from [11].

Interestingly, the addition of probabilities in the early cancer diagnosis model is done on a rule by rule basis, rather than as a separate module. From this point of view we already observe the "best of both worlds" situation: while purely probabilistic or statistical models of language provide little insight, as observed

in the discussion in [21], the incorporation of probabilities into rules that do provide cognitive insight by relating concepts links them into those insights, for mutual complementation.

As said, sometimes in a parsing process lexical, syntactic and semantic information must cooperate dynamically in order to zoom onto the precise meaning of a natural language sentence or discourse. In these cases we have found CHR to be particularly helpful, given that constraints springing from different grammar components can be conjured into the same CHR rule, which gets informed from all these sources simultaneously. A case in point is [15], which proposes a methodology that exploits this feature in order to develop a model of medical document de-identification.

De-identification is the process of automatic removal of all personally identifying Private Health Information (PHI) from medical records, while protecting the integrity of the data as much as possible [26]. In the current state of the art, although most of the performance metrics reported in every other paper, hits at least some 90% performance measure, most of the times, several restrictions to the input text and to the target output have been assumed. These mean we still need a human assistant to do at least the final scanning if not re-processing.

Our work on de-identification deals with privacy sensitive texts that are rich sources of research information by extracting the knowledge these texts represent and feeding them into a database, and by marking any sensitive fields syntactically for eventual removal by the system. The system takes into account what type of research will be conducted in order to protect identifying fields as needed. Thus researchers, instead of accessing the text, can query the database for the information they require, while having no access to specific identities. The hybrid nature of constraints involved (coming from different grammatical levels, or combining semi-structured with free text elements) can be elegantly handled by CHR's multi-headed rules. To illustrate the power of combining constraints from heterogeneous sources, lexical entries in the grammar for a hospital admissions application can glean information from an ontological component, resulting in lexical-semantic constraints that can deal with ambiguities such as:

```
enter(patient-X,hospital-Y). (as in "admitted into the hospital")
enter(patient-X,state-Y) (as in "entered into a comma")
```

The parser can then keep track of potential referents for pronouns and other referential terms through the use of assumptions. Interestingly, disambiguation and anaphora resolution can cooperate with each other: semantic types allow us to differentiate between a patient named Huntington and a disease named so; thus, further ensure the correct identification of a referent, as the following discourse and corresponding representations exemplify.

"Huntington entered the hospital on April 16, 2010. This patient should be tested for Huntington."

```
+entered(patient-id(huntington), hospital-id(universalcures),
```

```
date-id(16-04-2010)).
must-test-for(patient-P,disease-huntington)
```

Our parser’s anaphora resolution system will instantiate P with id(huntington) and correspondingly mark the relation “must-test-for” as an assumption. The explicit mention of the type (“patient”) in the subject of the second sentence serves as a corroboration to the anaphora resolution system that we are referring indeed to the Huntington typed as a patient, in the first sentence. If marked otherwise, the two types would not have matched. If the second sentence were “He should be tested for Huntington”, the type gleaned from the first sentence for this individual would simply carry over, together with his name, into the term representing it. Of course, even for humans there will be cases in which even context leaves us clueless, as in “Huntington won”. We are content if our proposed methodology allows us to deal with ambiguity with as much success as humans can.

5 Biological Languages

5.1 The languages of Nucleic Acid

Biological sequence analysis is resorting more and more to AI methods, given the astounding rate at which such information grew over the last decade. Old methodologies for processing it can no longer keep up with this rate of growth. AI methods such as logic programming and constraint reasoning have been coming to the rescue, generating a fascinating and interdisciplinary field. In particular, methodologies that pertain to the natural language processing field of AI are now being exploited to analyze biological sequences, which is uncovering similarities between the languages of molecular biology and human languages. [13], identifies some of the forms that tend to repeat in both human and molecular biology languages, and proposes a uniform treatment through CHR, regardless of the area of application.

Some of the forms found both in nucleic acid strings (made up of the “words” or nucleotides A, C, T, U and G) and in natural languages are relatively simple yet not necessarily easy to parse: e.g. *palindromes* (sequences that read the same from left-to-right or from right-to-left, as the Spanish sentence, modulo blank spaces: “Dabale arroz a la zorra el abad”, or as the DNA sequence A C C T G G T C C A). Their length can vary, and their position within in a string is unpredictable. *Tandem repeats* (where a substring repeats again right away) also appear in both types of languages, as “tut” does in “Tut, tut, it looks like rain”, or as C G A within the sequence C C A T C G A C G A U A). In human languages, repetition can appear literally or in more involved phenomena such as full conjunctive clauses, where the surface forms are not the same, but the structure repeats around some coordinating word like “and”, “or”, “but” (as in “Slowly but surely, ...”, where the tandem repeat is between two adverbs, and a mediating conjunction intervenes).

In addition to these basic structures, which can be found in linear sequences, pairings of nucleotides, which attract each other, form more complex structures, where the sequences fold into three dimensions: the nucleotide A tends to pair with T, and C with G. These are called *Watson-Crick* or *canonical base pairs*. These base pairings result in structures or motifs of a variety of forms, such as helix, hairpin loop, bulge loop and internal loop. One of the widely occurring complex structures in molecular biology is the *pseudoknot* which has been proved to play an important role for the functions of RNA. A simple pseudoknot is formed by pairing some of the bases in a hairpin loop that are supposed to stay unpaired, with bases outside the loop. If we draw for natural language sentences some of the links between, for instance, a clause's antecedent and the clause itself, we obtain similarly shaped figures.

5.2 Decoding nucleic acid through spoken language

High level code for analyzing nucleic acid strings can be written by computer specialists in reasonably useful and efficient ways. Still, it is the prerogative of computer specialists to write such code, even if in interaction with biology experts, and in some cases, of specialists in Artificial Intelligence. These are used to instructing computers to "think" logically and to conduct effective searches of large problem spaces by endowing their computer programs with reasoning capabilities, often based in executable incarnations of logic. The ability to encode such solutions in a suitable AI language is an acquired skill which requires extensive knowledge of the language, practice with writing programs in that language, and a lot of programming discipline and ingenuity. It moreover requires a suitable level of interdisciplinary communication skills in order to clearly capture the precise description of what is to be done, from the biologist who is interested in the results and for whom his own jargon is second nature. This involves the development of a common jargon or at least an understanding of the other's jargon for each of the disciplines involved. Not an easy task, but one in which good breakthroughs have been made and which advances at a quick pace.

In [9] we propose human language itself as the high level query language for decoding. We aim at an even higher level of interaction with computers- one in which biologists are given the software tools for commanding computers through their own human language such as English, to extract genetic information of their interest which is encoded in DNA strings. Ultimately we aim at doing away with the need to resort to a computer specialist- a task which seems formidable and perhaps is so in its full generality, but for which subtasks exist which are useful enough, quite impressive, and feasible.

Specifically, we extend a series of DNA decoding primitives written in CHR_G, such as those proposed in [4], into human language primitives which can then be used to automatically program the decoding of a nucleic acid string from a sentence that describes in plain language (English, French, etc.) what needs to be analysed within the string. Some efficiency concerns are tackled by appropriate constraints and thus remain invisible to the user, except in their effects.

Both the parser and the DNA decoding toolkit components of the system use CHR. Our approach allows for eager discarding of wrong lines of reasoning, as well as for paraphrases of a given question without ill-effects in the execution, and with consequent gain in the richness of the input accepted. As well, it permits a cooperative integration between both components, by allowing one of them to inform the other one through integrity constraints in CHR.

6 Cross-fertilizations between human and biological languages

6.1 A dual processing scheme for both human and biological languages

David Searls proved that the grammar of nucleic acid language is in fact non-deterministic and ambiguous and moreover not context-free [24]. These features are shared by so-called natural languages such as English.

Based upon these and other similarities, [13] proposes a model of human language processing, called Synalysis, built around CHR. Inspired by biological sequence replication and nucleotide bindings, this model can express and implement both analysis and synthesis in the same stroke. This is akin to biological mechanisms, such as DNA substring repair, in which a string is analysed while being synthesized elsewhere. The uses of Synalysis are exemplified around the language processing phenomenon of long distance dependencies, which also presents in molecular biology since it involves relating two substrings (of either human or biological language text) which might be arbitrarily far apart from each other. Our proposed model is suitable to those language processing frameworks known as *compositional*, where the representations obtained for the whole are composed out of partial representations obtained for the parts.

This research relies on CHR for the following reasons:

- ambiguity is inherently treated because all possibilities resulting from ambiguous input are expressed in the constraint store
- long-distance dependencies, including strings that repeat arbitrarily far apart, are easily conveyed through multi-headed rules
- in its grammatical version, CHR, we are spared from explicitly manipulating the input and output arguments and can specify context explicitly
- memoing, a well-established technique for human language processing which is also used in our dual model, is inherently available in CHR

Our research in [13] also shows that, while for the simpler of the forms we have identified as being common to both molecular biology and human language sequences, a uniform treatment through CHR is adequate in both disciplines, more complex forms might require the complement of heuristic rules. In our own research we have incorporated them through probabilities implemented in ad-hoc fashion, suiting our needs, because CHRiSM [25] was not yet available. It would be interesting to restate our results in terms of CHRiSM, since its present

availability may well prove to be an additional reason to favor CHR approaches to a unified processing scheme for both human and biological languages. In any case, we found that adding the needed probabilities to our CHR formulation was a straightforward enough task, giving further proof to our thesis that the two cultures can cooperate rather than compete.

6.2 Literary applications: an RNA-inspired analysis of poetry

The style in which an RNA molecule folds in space obeys laws of nucleotide binding and attraction which are encoded in its primary structure, that is, in the sequence of nucleotides conforming it. Natural language sentences can also be viewed as encodings for a structure in space- in this case, a parse tree- which exhibits relationships or bindings between different parts of the sentence. In [3], we presented a novel methodology –chrRNA– for addressing the bioinformatics problem of finding an RNA sequence which folds into a given structure. In [8] we explored the possibilities in adapting this methodology to the problem of parsing poems that follow specific stylistic trends, e.g. because they belong to the same author. Just as chrRNA involves a very simple grammar, which augmented by probabilities can lead to approximate but still useful solutions for a problem that has been proved to be NP-hard, and these probabilities encode the molecule’s ”style”, as it were, adapting our method to computational linguistics involves resorting to stylistic probabilities observed in a given author’s poetic production in order to aid in the parse of a given poem of the same author, or to aid in determining authorship itself. This methodology can also be applied to authorship determination.

As a simple example, consider the following sentence, adapted from Nicolas Guillen’s poem ”La Guitarra”: ”Dejo el borracho en su coche, dejo el cabaret sombrio”. This can be parsed into one sentence, which explicitly and in English would correspond to ”The drunkard left the sombre cabaret (by traveling) in his car”. Any Spanish reader would understand that the verb’s repetition (dejo=left) is for poetic effect, rather than a ”new” main verb. A machine analyzer, however, would recognize two sentences, corresponding either to: ”(Someone) left the drunkard in his car, and (the same person) left the sombre cabaret, or ”Someone left the drunkard in his car, and the sombre cabaret left”. The first interpretation is likely when one considers that implicit subjects are very common in Spanish (so any reasonable Spanish analyzer would conceive it); the second one is nonsensical for humans but plausible from syntax alone, and therefore, a fair candidate for a poetically uninformed parser. Note that while the state-of-the-art in parsing would allow us to choose between these two interpretations, perhaps by paying the price of including semantic type information to preclude non animated subjects such as ”the cabaret” for movement verbs such as ”dejo”, the interpretation as a single sentence with verb repetition would remain inaccessible, to the best of our knowledge, to state-of-the-art parsers, including those meant to analyze poetry. Algorithmic approaches to poetry have been around for a relatively long time, but they mostly focus on generating poetry by automated or semi-automated means (e.g. [18]), and as such, belong to the general field of

Electronic Writing. Automated poetry analysis, on the other hand, remains a bit more elusive, and concentrates on the more mechanizable subtasks, such as automated analysis of sound and meter.

As in the case of RNA design [3], we can encode probability values that will allow us to determine, in case of ambiguity, which possible analysis is more likely. Thus we can encode CHR rules with probabilities to the effect that when a verb is not the initial word in a sentence, the noun phrase that follows it is likely to be a direct object rather than a subject, so we can analyze it as the direct object of that verb. This rule will be used for instance for "dejo el cabaret sombrío".

Likewise, we can indicate through probabilities that initial verbs in Guillen's poetry are likely to appear before the subject noun phrase just for stylistic effect, as in "dejo el borracho" (literally, "left the drunkard").

7 Towards a model of humanistic investigation through CHR

We can distill, from all the discussed body of research, a series of features that make the CHR paradigm especially promising as potentially leading to a model of humanistic investigation:

- modularity allows for straightforward change for experimentation purposes
- concreteness is promoted by naturalness of bottom-up thinking
- flexibility, e.g. top-down and bottom-up strategies can coexist as needed
- potential for combining heterogeneous sources, which also serves for long distance dependency phenomena
- robustness: partial results are possible even if some data needed for a complete result is missing
- inherent treatments of ambiguity and memoing
- straightforward addition of probabilities, which are ubiquitously needed, in normal CHR or CHR_G rules
- recent availability of CHRiSM for further syntactic sugar

In addition, we note that as embedded in our language processing tools (Hyprolog, CHR_G), it presents the following further advantageous features:

- non-classical inference such as abduction and assumptions add flexibility and make it easy to explore what-if scenarios
- automatic handling of input and output arguments in the case of grammars.
- working store elements can come from a variety of disparate sources, and thus they lend themselves ideally for incorporating multi-agents that collaborate in tasks that require intelligent interactions with non-grammatical kinds of agents.

Crucial to humanistic investigation in any discipline is the formation of concepts in flexible enough a way that their properties can be enforced or relaxed as

needed. For this reason, we put forward that the CHR based paradigm of Concept Formation [12] may be appropriate in this respect. But as we have also seen, probabilities tend to play a major role in many of the applications described, as well as being important for natural language processing itself, an area which is relevant to all areas.

As proof of concept, we have abstracted, from recent different realizations of the linguistically inspired Concept Formation paradigm, a multi-agent model for Biological Concept Formation which can be considered as a computational metaphor for the (biological) mind, with direct executability implications [10]. Due to the generalized use of Constraint Handling Rules or their grammatical counterpart, we are able to integrate human language processing techniques into our approach which are not only useful for all types of concept formation but also allow us a smooth integration of human language processing agents, as well as their interactions with the knowledge base agents. Another interesting feature of our proposal is its robustness: due to the capability of relaxing some of the properties involved in concept formation, results that can be useful are provided even in the absence of all the information "necessary" to form the concepts in question.

Concept formation rules are applicable to many other AI and cognitive problems as well, most notably, those involving the need to reason with incomplete or incorrect concepts.

Another important facet of humanistic investigation is parsing spoken language itself, which will be important in any discipline. The parsing model needed for Language Intelligence as we understand it (i.e., much beyond keywords and syntactic variants) must satisfy three main requirements: ability to decode text into knowledge bases, flexibility to accommodate the imperfections and imprecision typical of spontaneous human language use, while exploiting its rich expressive power to good advantage, and good potential to blend in, and cooperate with, semantic web technologies. Adapting the new family of Abductive Grammars [6, 16] holds great promise in this respect, because of their built-in ability to construct knowledge bases from language sentences. As well, our constraint-based rendition of Property Grammars [11]) holds great promise because of their focus on yielding useful results even for imperfect input (involving noise, incorrect input, and incomplete input).

8 Concluding Thoughts

Just as our root discipline – Philosophy – needed to slowly separate into a myriad of disciplines and sub-disciplines in order to develop methods specific to each, to achieve depth, etc., we believe we are now at a time in which an inverse process of integration needs to happen: in specializing, some disciplines have become unnaturally disconnected from others or from the whole, with the result that the broad view of the forest is sometimes lost, and that parallels that could be exploited cannot even be seen. A reconnection from the more mature present standpoints of these different branches seems in order and in any case,

is simply happening. We hope to have shown, in a very modest way and for just a few of the disciplines needing this, that the CHR paradigm can constitute a good pivot around which to perform the needed reconnection, and that whereas constraint programming seems more and more focussed on solving "classic" OR problems and benchmarking, it is not mostly about gaining 1 ms on instance i of problem p , but is also a powerful descriptive tool for addressing interdisciplinary applications with the mighty advantage of direct executability.

Acknowledgement

Support from NSERC's Discovery Grant 31611024 is gratefully acknowledged. I also thank the anonymous referees for very useful comments on this article's first draft.

References

1. Blache, P.: Property Grammars, a Fully Constraint-Based Theory. In: Constraint Solving and Language Processing, H. Christiansen et al. (eds), LNAI 3438, Springer (2005)
2. Barranco-Mendoza, A. Persaoud, D.R., Dahl, V.: A Property-Based Model for Lung Cancer Diagnosis. Poster, RECOMB 2004, San Diego (2004) 27–31
3. Bavarian, M. , Dahl, V. : Constraint-Based Methods for Biological Sequence Analysis. Journal of Universal Computing Science (2006) 12(11) 1500–1520
4. Bel Enguix, G., Jimenez-Lopez, M.D., and Dahl, V.: Mining Linguistics and Molecular Biology Texts through Specialized Concept Formation. Poster, NLPCS09
5. Christiansen, H. and Dahl, V.: HYPROLOG: a New Logic Programming Language with Assumptions and Abduction. LNCS 3668: 159-173 (2005)
6. Christiansen, H., Dahl, V.: Abductive Logic Grammars. In: Ono, H., Kanazawa, M., de Queiroz, R.J.B. (eds.) WoLLIC 2009. LNCS, vol. 5514, pp. 170–181. Springer, Heidelberg (2009)
7. Christiansen, H.: CHR Grammars. Journal on Theory and Practice of Logic Programming. 5, 467–501. (2005)
8. Dahl, V., Jimenez-Lopez, M.D., and Perriquet, O. Poetic RNA: Adapting RNA Design Methods to the Analysis of Poetry. IN: PAAMS 2010, Vol. 2, pp. 403-410, Springer Verlag, ISBN 978-3-642-12383-2. (2010)
9. Dahl, V.: Decoding Nucleic Acid Strings through Human Language. Bel-Enguix, G., Jiménez-López, M.D. (eds.) Language as a Complex System: Interdisciplinary Approaches. Cambridge Scholars Publishing (2010)
10. Dahl, V., Barahona, P., Bel-Enguix, G., Kriphal L.: Biological Concept Formation Grammars- A Flexible, Multiagent Linguistic Tool for Biological Processes. LAMAS (2010)
11. Dahl, V., Blache, P.: Directly Executable Constraint Based Grammars. In: Journées Francophones de Programmation en Logique avec Contraintes, Angers, France (2004)
12. Dahl, V., Voll, K.: Concept Formation Rules: An Executable Cognitive Model of Knowledge Construction. In: 1st International Workshop on Natural Language Understanding and Cognitive Sciences. Porto, Portugal (2004)

13. Dahl, V., Maharshak, E.: DNA Replication as a Model for Computational Linguistics. In J. Mira et al. (Eds.): IWINAC09 (Best Paper Award). LNCS, vol. 5601, pp. 346-355. Springer-Verlag, Heidelberg (2009)
14. Dahl, V., Gu, B.: Semantic Property Grammars for Knowledge Extraction from Biomedical Text. In: 22nd International Conference on Logic Programming (2006)
15. Dahl, V., Saghaei, S., Schulte, O.: Parsing Medical Text into De-identified Databases. In: 1st International Workshop on AI Methods for Interdisciplinary Research in Language and Biology, Rome (2011)
16. Dahl, V.: From Speech to Knowledge. In: Paziienza, M.T. (Ed.) Information Extraction: towards scalable, adaptable systems. Springer, LNAI 1714, Springer, pp. 49-75 (1999)
17. Dahl, V. and Tarau, P.: Assumptive Logic Programming, Proc. ASAI'04, Cordoba (2004)
18. Mendelowitz, E.: Drafting poems: inverted potentialities. In: International Multimedia Conference, Proceedings of the 14th Annual ACM International Conference on Multimedia, pp. 1047-1048. Santa Barbara (2006).
19. Fruhwirth, T.W.: Constraint Handling Rules. Cambridge University Press, ISBN 9780521877763, 2009.
20. Tarau, P. (2011) The BinProlog Experience: Architecture and Implementation Choices for Continuation Passing Prolog and First-Class Logic Engines. CoRR abs.1102.1178.
21. Norvig, P.: On Chomsky and the Two Cultures of Statistical Learning. <http://norvig.com/chomsky.html>.
22. Pollard C. & I. Sag (1994), *Head-driven Phrase Structure Grammars*, CSLI, Chicago University Press.
23. Prince A. & Smolensky P. (1993), *Optimality Theory: Constraint Interaction in Generative Grammars*, Technical Report RUCCS TR-2, Rutgers Center for Cognitive Science.
24. Searls, D.: The Language of Genes. *Nature*, 420, 211–217 (2002)
25. Jon Sneyers, J., Meert, W., Vennekens, J. CHRiSM: Chance Rules induce Statistical Models. In: Proc. CHR'09 Workshop, Pasadena, CA (2009)
26. Uzuner, O., Luo, Y., and Szolovits, P.: Evaluating the state-of-the-art in automatic deidentification. *Journal of the American Medical Informatics Association*, 14(5):55063 (2007)