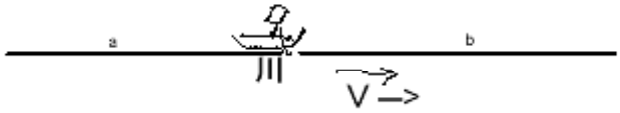
青葉設計書　　2013/04/25（木）

【問題】

川が時速 3Km で地点 a を通り地点 b を抜けて下流にながれている。ボートで a から b まで往復漕いでいく。往きは５分かかった。帰りは１０分かかった。a と b との距離はいくらか。

ただし、ボートの静水に対して漕ぐ速度は一定であるとする。

この問題にイメージは次のようになるでしょう。



文章の意味解析結果はこんな所でしょうか。

(流れる：move,agent_case,川,from_case,地点 a,to_case,地点 b)

^(BE,agent_case,move,attribute,時速 3Km);

(漕いでいく：move,agent_case,私,from_case,a,to_case,b,use_case,ボート)

^(BE,agent_case,move,attribute,往復);

(かかる：use;agent_case,私, object_case,time,through_case,往路)

^(BE,agent_case,time,attribute,5 分);

(かかる：use;agent_case,私, object_case,time,through_case,帰路)

^(BE,agent_case,time,attribute,10 分);

(quetion,object_case,距離)^(BE,agent_case,距離,attribute,L)

^(BE,agent_case,L,from_case,a,to_case,b);

問題文章を式を連想する意味ネットワークに落とすには、文章の文脈を把握していかねばなりません。文脈が主で、各文は従の関係で、問題文章を通して、述べられていない情報を埋め込み、同じものを指す名詞は何かなどを解析して、完全な文脈を把握しなくてはなりません。この文章では、「地点 a」と「a」は同じと判断します。「往路」は「from a to b」で、「帰路」は「from b to a」でしょう。

また、文脈は行動すら、要素に分けて把握します。この例文では問題になりませんが、「A は B に C を渡す」ですと、「C belong A」「C move from A to B」「C belong B」と分析

していくことが重要です。物理学の問題では、この要素と捉えて式を立てていくことになるでしょう。

意味ネットワークを導出するときに捉えておきたいことがあります。それは、場面イメージの把握のカテゴリーです。カテゴリーは次のようなタップルになります。

（場、土台、時間、アクター、配置（格)、変化）

「場」はベクトル場とか重力場とかの空間にベクトルが分布している状況です。

「土台」は事象が起きている場所です。

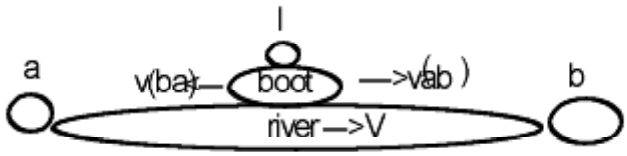
「時間」は事象が起きている時点、時間です。土台と時間があって、ステージとなります。

「アクター」は登場人物、もの等、オブジェクトとして捉えるべきもの全てです。

「配置」はアクターの間の２項関係です。

「変化」は運動とか行動とかです。コマンドになります。

こうして捉えた問題文章の意味ネットワークは次のようになるでしょう。



円はオブジェクトで円の接触はオブジェクトの接触で、矢印は場です。

【解法】

意味ネットワークを基盤として、次のようなプロダクションシステムで式と変数を立てていきます。

（１）目標設定

(calucurate,object_case,L)^(BE,agent_case,L,attribute,距離)

^(BE,agent_case,距離,between_case,a,and,b);

(make,object_case,formula)^(BE,agent_case,formula,attribute,complete);

(BE,agent_case,FORMURASET,attribute,blackbord);

(BE,agent_case,BBX,attribute,blackbord);

(BE,agent_case,BBF,attribute,blackbord);

（２）プロダクションシステム

(find,object_case,fomura,blackbord_case,FX)^(add,object_case,FX,to_case,FOMURASET)
^(BE,agent_case,formura,attribute,new)
^(count,object_case,valuable,blackbord_case,X)^(add,object_case,X,to_case,BBX)
^(count,object_case,fomula,blackbord_case,F)^(add,object_case,F,to_case,BBF);
(stop;true_case,T)^(BE,agent_case,BBX,agent_case,BBF,attribute,equal)
^(include,agent_case,FORMURASET,object_case,L);

プロダクションシステムのデータによって、ＦＯＭＵＲＡＳＥＴに式を設定していき、これからあとは、Ｌを求める方程式を解くだけです。数式処理の問題となります。数式処理も、コマンドシステムの実現していくだけです。

そもそもの式ですが、distance=verocity*time;などは意味ネットワークの場の情報として持っていることにします。ベクトル合成の式もそうです。式を探すのは、この場のなかの式情報を検索することで実現します。式は意味記号で表現していて、立式には変数をこの意味記号の代わりに利用します。

ここで、意味ネットワークですが、２つのデータタイプが必要のようです。それは、ノードのネットワークとネットワークのノードを記述する２つの情報が必要なことが理由です。角度は２つの隣接する線分とノードとで記述されます。ノードの記述は、

(BE,agent_case,ANGLE12,between,LINE1,and,LINE2);

などと表現する、オブジェクトと格の対の連鎖です。ノードは基本的にオブジェクト一つで表現します。

青葉では、オブジェクトを Node クラスで表現し、その持つデータは、

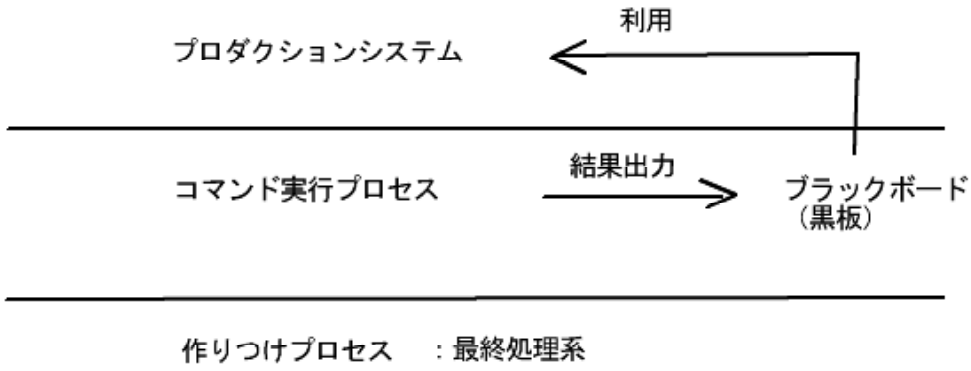
（１）ID・・・・・・・ローカル名
（２）NAME・・・・・グローバル名
（３）ＴＹＰＥ・・・・オブジェクトのタイプ
（４）ＷＯＲＤ・・・・情報名

です。

これを格情報と一緒に持つ ObjectReferent クラスを設けています。そうして、ノードを表現するのは、ObjectReferentSentence で、ObjectReferent の連鎖で完全にノードを表記します。

意味ネットワークは、各ノードの隣接関係をポインタで保持する、ObjectReferentSentenceNetNode というクラスを起こして、そのネットワークで表現しています。

プログラムの構造は、一対一対応のプロセスとか、集合論プロセスとか、作りつけのプロセスクラスと、コマンドを実行するプロセスから成っています。コマンドを実行するプロセスは最終的に作りつけプロセス群に制御を落としていき実行されます。この辺は、機械学習を想定しています。そうして、それらの上位にプロダクションシステムの知識があるのです。プロダクションシステムは処理結果を黒板システムに格納します。基本的に黒板は、一つのコマンドの実行時にクリアしますが、結果をセーブするコマンドだけはクリアせず、結果をスタックしていきます。



こんなところが青葉の概略設計です。

おわり。